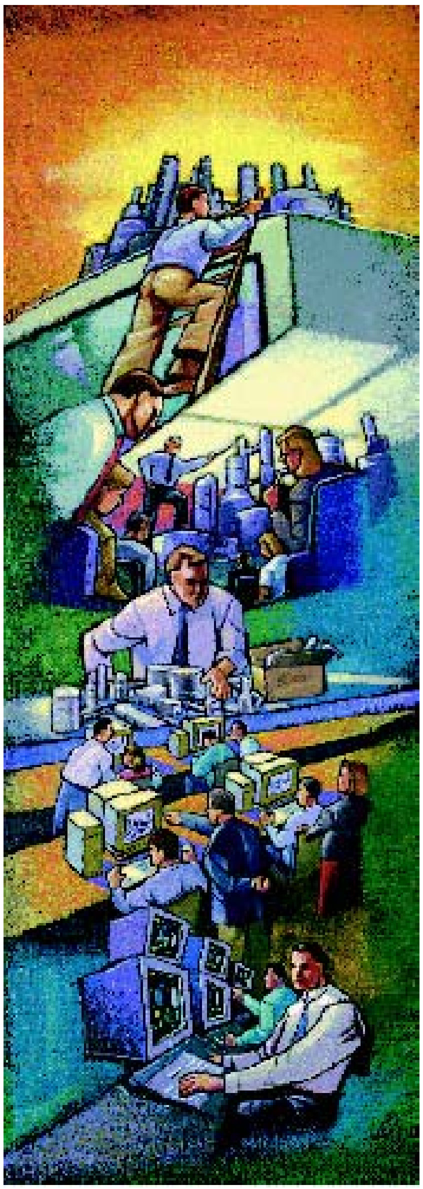
SKC 5771
Process Control Laboratory
Student's Manual

Second Edition and Revised



MOHD KAMARUDDIN ABD HAMID
with
AHMAD KAMAL BADERON
KHAIRIYAH MOHD YUSOF
MIMI HARYANI HASSIM
Process Control & Safety Group

Universiti Teknologi Malaysia

Faculty of Chemical and Natural Resources Engineering

SKC 5771
Process Control Laboratory

Student's Manual

Second Edition and Revised

©2005 by PCS UTM

SKC 5771
Process Control Laboratory
Student's Manual

Second Edition and Revised

MOHD KAMARUDDIN ABD HAMID
with
AHMAD KAMAL BADERON
KHAIRIYAH MOHD YUSOF
MIMI HARYANI HASSIM
Process Control & Safety Group

Universiti Teknologi Malaysia

Faculty of Chemical and Natural Resources Engineering
July 2005

©2005 by PCS UTM

Preface

- Course Objectives
- Lab Experiments
- Groups/Teams
- Safety
- Students Performance Evaluation

Course Objectives

By the end of the course, students should be able to:

- Develop a constant awareness of safety in the laboratory so that all laboratory work is carried out in a safe manner.
- Develop the ability to carry out experimental investigations of pilot-scale processes including:
  - Creating equipment diagrams and comprehensive safe operating procedures for various unit operations.
  - Determining a specific set of experimental objectives when given a vague objective by the instructor.
  - Planning an experimental strategy for collecting the appropriate data.
  - Obtaining the experimental data required to satisfy the objectives.
  - Gaining competency in analyzing experimental data and in comparing the results to data and theories in the literature.
  - Reporting the results of the experiment in a concise, well written, well documented written report.
  - Presenting the results in a professional, oral presentation using available technological resources.
- Develop the ability to work in a team by:
  - Actively participating as a member of a professional group.
  - Leading a peer group of professional chemical engineers.
  - Managing conflicts within the team as they arise.
- Develop confidence through the application of previously acquired knowledge of unit operations, chemical reactions, process safety, and process control.
- Learn to apply software tools typically used by process control professionals.

Lab Experiments

SKC 5771 Lab consists of 8 experiments designed to support the concepts described in lecture. Descriptions for each experiment are contained in the *SKC 5771 Process Control Laboratory:*

©2005 by PCS UTM

Student's Manual. Students are expected to be thoroughly acquainted with the descriptions prior to performing each experiment. The titles of these experiments are summarised below:

Experiment 1: Level Control Pilot Plant

Experiment 2: Modelling Heat Exchanger Pilot Plant

Experiment 3: Heat Exchanger Temperature Control Pilot Plant

Experiment 4: Flow Control Pilot Plant

Experiment 5: Steady State Feedback Control

Experiment 6: Distillation Controller Tuning

Experiment 7: Programmable Logic Controller (PLC)

Experiment 8: Dynamic Simulation in SIMULINK

Groups/Teams

For lab work, groups of maximum five (5) will be assigned randomly during the semester for each experiment. Each lab group will have a Team Leader, a Theorist, and an Experimentalist. Typical roles are as follows:

Team Leader: responsible for organizing group, conducting a safety analysis; outlining project and report; writes introduction, objectives & executive summary, answers questions assigned with each lab, and prepares recommendations with the experimentalist.

Theorist: responsible for relating experiment to engineering theory, including equations; writes background and reference sections. Assists in preparing analysis spreadsheets.

Experimentalist: determines data to collect (with theorist), prepares procedures and experimental parameters; sets-up experiment; writes materials and procedures sections of report, including diagrams/sketches of experimental set-up, where appropriate, and contributes to recommendations section of report. Assists in data analysis.

Analysis: (all members) prepares analysis spreadsheet; prepares & analyzes results; writes results and discussion section of report.

Safety

Following sound safety procedures while working in the lab is **mandatory**. Students failing to meet the required eye wear and dress codes, or who engage in negligent behaviour, will be dismissed from the laboratory without the opportunity for a make-up lab. Likewise, no food or drink is allowed in any of the laboratories. Each team must show a signed statement (Pre-lab) verifying that they have read and that they understand the manual before they will be permitted to begin lab work.

©2005 by PCS UTM

Students Performance Evaluation

Each student will perform each of the eight standard experiments. The selected random groups will be posted. You will do the following for each standard experiment:

1. Review the problem statement and laboratory manual.
2. Perform experiment on the day assigned.
3. Prepare as a group analyzed data tables (spreadsheet), and draft figures, due at the end of the experiment and graded as part of your draft report.
4. Prepare as an individual 3 pages handwritten draft of the final report (Introduction, Theory, Apparatus and Procedure, Safety, Reference cited) due two days after the lab. Attached your group results.
5. Prepare a final report (group) in memo format - due two weeks later.

For each experiment:
Handwritten draft report (individual) – 20%
Tabular analyzed data, draft figures (group) – 10%
Final report (group) – 70%

Total point breakdown is as follows:

Lab Experiment 1 :	100
Lab Experiment 2 :	100
Lab Experiment 3 :	100
Lab Experiment 4 :	100
Lab Experiment 5 :	100
Lab Experiment 6 :	100
Lab Experiment 7 :	100
Lab Experiment 8 :	100
Oral Presentation :	100
Peer Rating :	100
Total Lab Points:	**1000**

Students will do the **Peer Rating** at the end of the semester during **Oral Presentation**. Information about oral presentation will be given from time to time.

Note: No late reports are accepted.
All experiments must be completed with a grade of 60 or better to pass the course. You will ask to rewrite the report if you make a lower grade.

©2005 by PCS UTM

Contents

©2005 by PCS UTM

©2005 by PCS UTM

©2005 by PCS UTM

1

Level Control Pilot Plant

1.1 Simple PID Level Control (LIC-221-CD600)-Self-Regulating

Level process control is a control system via computer system to monitor the level of liquid in tanks and vessels and to enable automatic adjustments to the level of liquid through mechanical means. Its applications are widely being utilized in plant control and automation throughout the world.

The basic mechanism that is used in real industry is demonstrated in the *SimExpert Level Control Pilot Plant.* This Pilot Plant represents a typical Boiler Drum Level Process found in most industries. Feedwater is pumped from a feedwater tank into the Boiler Drum. Drum Water level is controlled via a feedwater control valve. Two types of level processes have been designed into the pilot plant, namely an Integrating Process and a Self-regulating process. Simple PID Control of level and Cascade Level Control with Flow control loop are provided for each of the two processes above. Level measurement based on both Open Tank and Closed Tank techniques are provided. The control valve, being one of the most important element in the control loop, has been careful selection of with proper sizing and valve characteristics. Control valve trim material is made of Stainless Steel for durability. Flow measurement based on common measurement technique for liquid, such as Differential Pressure (Orifice plate)

©2005 by PCS UTM

flowmeter, magnetic flowmeter, or Vortex flowmeter is provided depending on user requirements.

1.1.1 Objective

This experiment is divided into Four (4) sections:

- Section A: **Identify the Process,**
- Section B: **Open Loop Controller Tuning,**
- Section C: **Loop Controller Tuning,** and
- Section D: **Performance Test for the Control Loop.**

1.1.2 Results

The student is expected to discuss a report on the findings about the process and make a comparison of the proposed tuning methods.

1.2 Preparation

STEP	ACTION	REMARKS
1	Ensure that all Utility Services are ready (Switch on Power Supply to Control Panel).	
2	Open the hand valve HV211 and fill up the Feed-water Tank VE210 with water to the high level.	
3	In Local Control Panel, switch the Experiment Selector Switch to "**Level**".	
4	At the CD600 Controller font panel, select loop 1 by pressing <LP> until the Tag "LIC221" show on display.	
5	At the CD600 Controller font panel (Control Loop 1), set the controller to Manual mode	Press "A/M" button
6	Manually open the control valve FY221 to 50%.	Set MV to 50 by pressing Manual Actuation button.
7	Start the pump P213. Pump water into the Boiler VE220 until it is about half-full.	Push P213 Start push button.
8	Record down the flow reading on the flow transmitter FT221.	
9	Open the hand valve HV214 fully and monitor the water level.	
10	If needed, apply pressure to the VE220 as to balance up the water level.	By monitoring the pressure reading of PI220 and comparing the flow of FT221 and FI221.

1.3 Section A: Process Identification

1.3.1 Start Experiment

STEP	ACTION	REMARKS
1	At the CD600 Controller font panel (Control Loop 1), manually open control valve FY221 to the stable value obtains from the preparation.	Set MV by pressing Manual Actuation button.
2	Open the hand valve HV214 fully. Apply the designed pressure.	By referring PI220.
3	Start the Pumps P213 via Local Panel.	Push P213 Start push button.
4	Observe Process Value (PV) from the CD600 Controller Display and wait until it has to a constant value.	In this case, the PV is the Water Level.
5	At the CD600 controller font panel (Control Loop 1), manually apply a step change to the control valve FY221 by an additional 10%.	By pressing Manual Actuation button.
6	Observe Process Value (PV) from the CD600 controller Display and wait until it has to a constant value.	This is the process Reaction curve
7	Tear out the chart from the recorder.	This is the process Reaction curve
8	Stop the Pumps P213 and the pressure supply.	Push P213 Stop Push button.

1.3.2 Results Analysis

1	Compare the level curve with a set of expected process Reaction Curve provided in **Appendix A** of this manual.	
2	Identify the process response with the corresponding Reaction Curve	
3	Make several measurements as per the **Reaction Curve** chart.	Refer Appendix A-1
4	Sketch a Block Diagram to represent the process and Describe the Characteristic of this Process.	Dead time, Capacity/ Rate of rise, Time Constant. Noise, etc
5	Suggest the suitable Control Modes for a PID Feedback Control Loop for regulating of the process.	E.g. P, I, D settings

1.4 Section B: Open Loop Tuning Method

1.4.1 Tabulation and Analysis of Results

STEP	ACTION	REMARKS
1	Using the printed graph obtained from Section above (process Analysis) above, measure and tabulates the value as required. Refer to **Appendix B, Table B-1**.	Note: dB_u and dM are change from the 1st stable output to the 2nd.
2	Base on the equations for Open Loop Tuning, calculate the required controller tuning parameters. Refer to **Appendix C, Table C-1**	
3	At the CD600 Control font panel (Control Loop 1), set the calculated controller tuning parameters to controller.	
4	You are now ready to test the performance of the Control Loop. Proceed to **Section D** to continue.	

1.5 Section C: Closed Loop Tuning Method

1.5.1 Start Experiment

STEP	ACTION	REMARKS
1	At the CD600 Controller font panel (Control Loop 1), set the controller to Manual Mode and Close the valve.	
2	Set the Controller Gain to 0.5, the Integral time to 200, and the Derivative time to 0	
3	At the CD600 Controller font panel (Control Loop 1), adjust the Set Point (SP) to 50%.	
4	Open the hand valve HV214 fully. Apply the designed pressure.	By referring PI220.
5	Start the Pump P213 via local Panel.	Push P213 start push button.
6	Set the Controller to Manual Mode, Slowly open the control valve FY221, to bring the process value (PV) to almost equal to the set point	Adjust through MV
7	Observe process Value (PV) from the CD600 controller Display and wait until it has stabilized to a constant value.	
8	Set the Controller to Auto Mode	
9	Wait for the Process Value (PV) to stabilize.	
10	Make a Small Step Change to the Set Point (increase the set point by 10%)	

©2005 by PCS UTM

11	Observe Process Value (PV) from the Display or the Process Value of CD600 controller, if the PV response is not oscillatory, double the controller gain value until it becomes oscillatory.	
12	If the PV response is oscillatory, observe whether the magnitude of PV is increasing or decreasing. If it is increasing, reduce the controller gain by 1.5 times. If the PV is decreasing, increase the controller gain by 1.5 times. Aim to obtain an oscillatory response with almost constant amplitude.	
13	When Constant Amplitude Oscillation is achieved, allow up to 3 or more oscillation cycles to be recorded.	
14	Stop Pump P213, and then set the controller (Control Loop 1) to manual mode.	
15	Tear out the chart from the Recorder.	

1.5.2 Results Analysis

1	Using the printed graph obtained from Section above, measure and tabulates the values as required. Refer to **Appendix B, Table B-2**.	
2	Base on the equations for Closed Loop Tuning, calculate the required controller tuning parameters. Refer to **Appendix C, Table C-2**.	
3	At the CD600 Controller font panel (Loop 3), set the calculated controller tuning parameters to controller.	
4	You are now ready to test the performance of the Control Loop. Proceed to **Section D** to continue.	

1.6 Section D: Control Loop Performance Test

1.6.1 Start Experiment

STEP	ACTION	REMARKS
1	At the CD600 Controller (Control Loop 1), set its P, I, D parameters obtained from the previous.	
2	Adjust the Set Point (SP) of LIC221 to 50%	
3	Set the controller to Manual Mode	
4	Open the hand valve HV214 fully. Apply the designed pressure.	By referring PI220.

5	Start the Pump P213 via local panel.	Push P213 Start Push button.
6	Adjust the controller output (MV) until its process value (PV) is close to its set point (SP)	
7	Set the Controller to "Auto Mode"	
8	Observe Process Value (PV) from the CD600 Controller Display until it is reasonable stable.	
9	Increase the Controller Set Point (SP) by making a step change of between 10% to 20%.	
10	Observe Process Value (PV) from the CD600 Controller Display until it is reasonable stable.	
11	Tear out the chart from Recorder. Look for some typical response characteristics.	
12	Capture the important process responses from chart.	
13	Stop the Pump P213, the pressure supply and then set the controller to manual mode.	

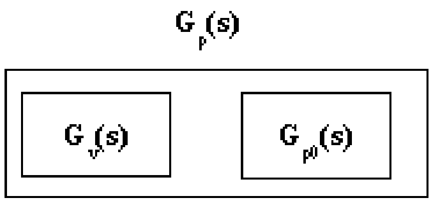
1.6.2 Tabulation and Results

1	Using the printed graph obtained from Section D.1 above, measure and tabulates the values as required. Refer to **Appendix B, Table B-3**.	
2	Describe the Characteristic of the Process response.	
3	Discuss the functions of each controller tuning parameters, P, I, D.	
4	Suggest any improvements to the process control loop and its total error.	

1.7 Appendices

1.7.1 Appendix A

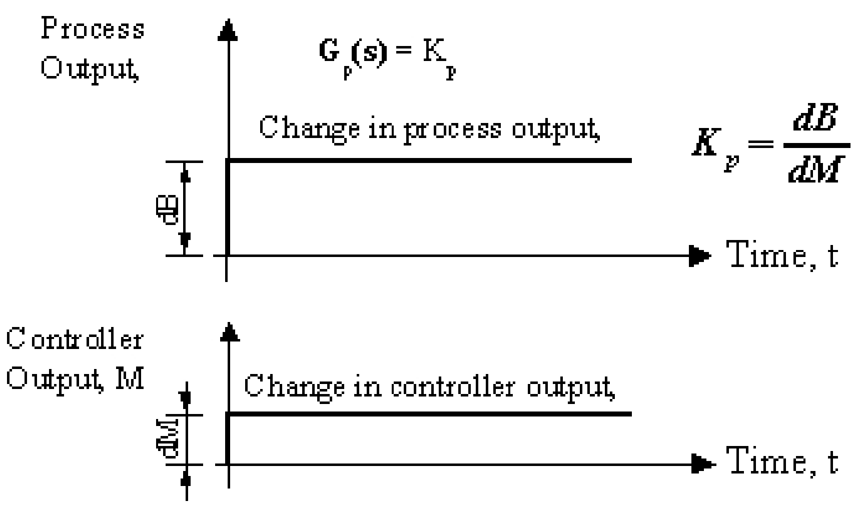
Process Block Diagram



$$G_p(s) = \frac{[\text{Change of Process Output}]}{[\text{Change of Process Input}]}$$

Fig A-1: Process Model inclusive of Control Valve

Process Reaction Curve 1



$$K_p = \frac{dB}{dM}$$

Fig A-2: Resistive Process Reaction Curve

Process Reaction Curve 2

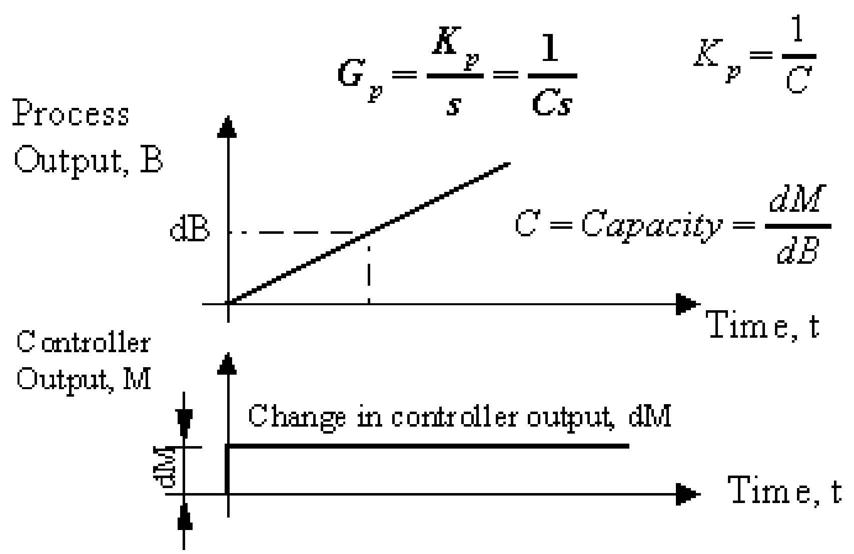


Fig A-3: Integratnig Process Reaction Curve

Process Reaction Curve 3

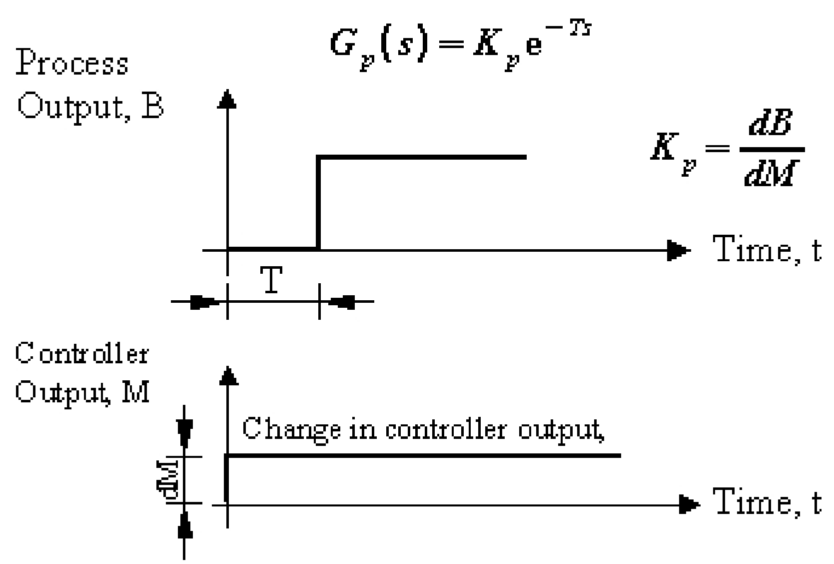


Fig A-4: Dead-time Process Reaction Curve

Process Reaction Curve 4

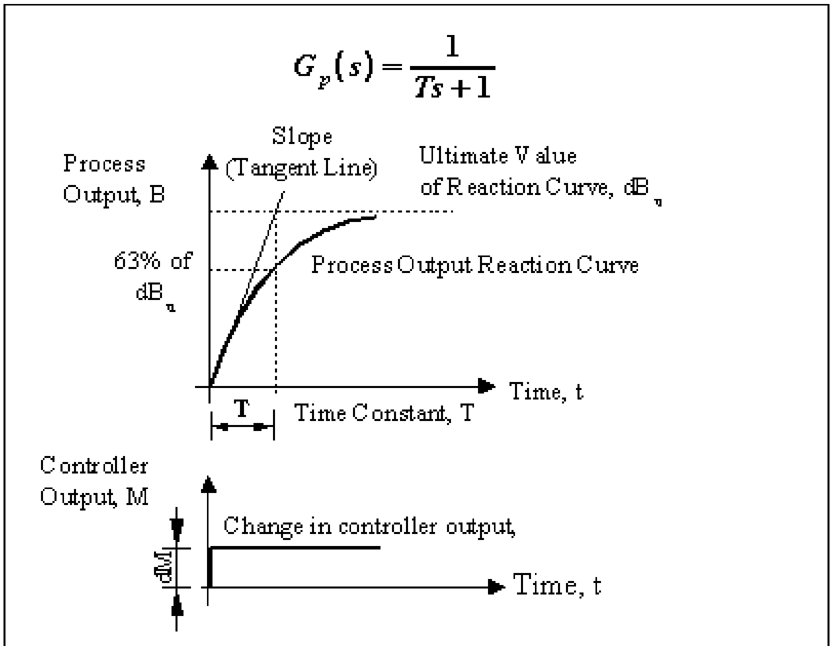


Fig A-5: First Order Process Reaction Curve

Process Reaction Curve 5

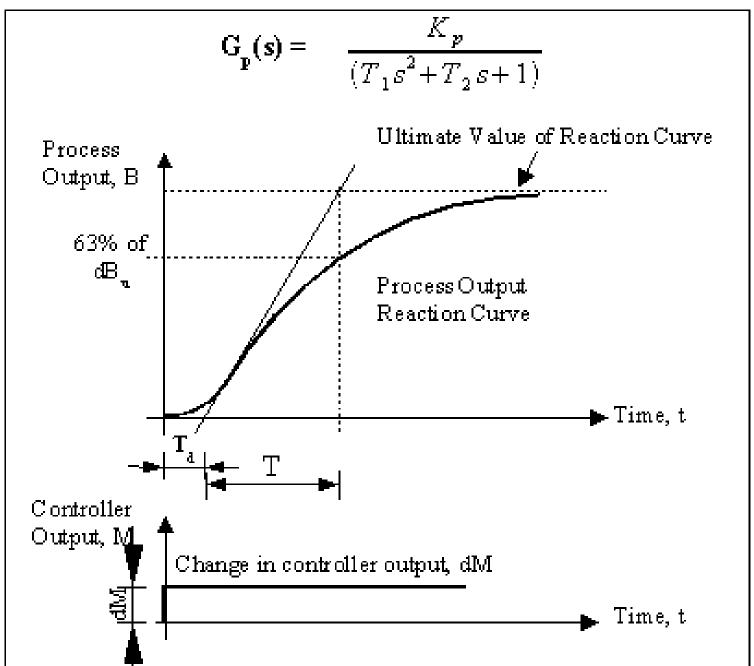


Fig A-6: Second Order Process Reaction Curve

1.7.2 Appendix B

Table B.1: Results for Open Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
Change in Manipulated Variable, dM				
Change in Ultimate Value, dB_u				
Slope S				
Apparent Dead Time, T_d				
Calculations:				
Apparent Time Constant $T = dB_u/S$				
Steady State Process Gain $K_P = dB_u/dM$				
$R = T_d/T$				
Tuning Parameters:				
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.2: Results for Closed Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
$K_{u,}$ Ultimate Controller Gain				
Time for 3 Oscillation periods, or more				
CALCULATIONS:				
Ultimate Time T_{u} (Time for one Oscillation period)				
Tuning Parameters:	**Test 1**	**Test 2**	**Test 3**	**Average**
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.3a: Results for Flow Compensation Calculation FZ210

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT210 , m^3/h						
Corresponding Line Pressure PT210, barg						
Corresponding Line Temperature TT210, degC						
Compensated Flow from DCS, FZ210, scmh						
Constants: q_{Base} , scmh	32	32	32	32	32	32
Constants: p_{Base} , barg	5	5	5	5	5	5
Constants: t_{Base} , degC	30	30	30	30	30	30
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_F , scmh						
% Error $\frac{FZ210 - q_F}{FZ210} x 100$, %						

Table B.3b: Results for Flow Compensation Calculation FZ211

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT211, m^3/h						
Corresponding Line Pressure PT211, barg						
Corresponding Line Temperature TT211, degC						
Compensated Flow from DCS, FZ211, scmh						
Constants: q_{fnom}, scmh	32	32	32	32	32	32
Constants: p_{fnom}, barg	5	5	5	5	5	5
Constants: t_{fnom}, degC	120	120	120	120	120	120
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_r, scmh						
% Error $\frac{FZ210 - q_r}{FZ210} x\,100$						
Flow comparison, % Error $\frac{FZ211 - FZ210}{FZ210} x\,100$						

Table B.5: Results for Control Valve Characterisation

Measurement	Pont 1	Pont 2	Pont 3	Pont 4	Pont 5	Average
Opening of FCV211 (%)						
Source Pressure, PT202 (bar)						
Inlet Pressure, PT211 (bar)						
Outlet Pressure, PT212 (bar)						
Exit Pressure, p_e (bar)	0	0	0	0	0	
Compensated Flow FZ211 (scmh)						
Characteristic Constant, a						
CALCULATIONS:						
Pressure across valve when it is 100% open $dp_{100} = PT211 - PT212$						
$K = q_{100}/dp_{100}$						
Total System Pressure head $dp_{sys} = PT202 - p_e$						
Max. Pressure across valve $dp_{max} = PT211 - PT212$ at minimum flow ie. When valve is 10% open, (bar)						
Min. Pressure across valve $dp_{min} = PT211 - PT212$ at maximum flow ie. When valve is 100% open, (bar)						
Relative Pressure drop across valve $R_v = dp_{max}/dp_{min}$						

Note: q_{100} is the volumetric flow at standard conditions when control valve is 100% open, in scmh

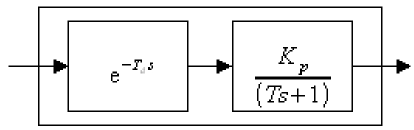
dp_{100} is the differential pressure across the control valve when it is 100% open, in bar

1.7.3 Appendix C

Calculation of Open Loop Tuning Parameters

Cohen-Coon Rules

The Cohen & Coon tuning rule assumes that the S-shaped Process Reaction curve can be approximated by a process model consisting of a First order lag and a Dead Time.



$$G_p(s) = \frac{[K_p e^{-T_d s}]}{(Ts+1)}$$

Fig C.1 Approximation to S-shaped Process Reaction Curve

Table C.1: Open Loop Tuning – Controller Parameter Calculations

Control Modes	Calculation	(R = T_L/T)
P only	$[\frac{1}{R.Kp}][1+\frac{R}{3}]$	K_c =
P + I	$[\frac{1}{R.Kp}][\frac{9}{10}+\frac{R}{12}]$	K_c =
	$Td.\frac{(32+6R)}{(13+8R)}$	T_I =
P + I + D	$[\frac{1}{R.Kp}][\frac{4}{3}+\frac{R}{4}]$	K_c =
	$Td.\frac{(32+6R)}{(13+8R)}$	T_I =
	$Td.\frac{4}{(11+2R)}$	T_D =

©2005 by PCS UTM

Calculation of Closed Loop Tuning Parameters

Ziegler-Nichols Rules

Table C.2: Closed Loop Tuning – Controller Parameter Calculations

Control Modes	Kc
P only	$K_c = 0.5K_u$
P + I	$K_c = 0.45K_u$
	$T_I = T_u/1.2$
P + I + D	$K_c = 0.6K_u$
	$T_I = T_u/2$
	$T_D = T_u/8$

2

Modelling Heat Exchanger Pilot Plant

2.1 Modelling Heat Exchanger Pilot Plant

Heat Exchanger-Temperature Control Pilot Plant is a control system via computer to monitor the heat exchange between hot and cool liquid inside each tank and vessels. Each process can be adjust or control using the computer either by user or itself. Its applications are widely being utilized in plant control and automation throughout the world.

The basic mechanism that is used in real industry is demonstrated in the *SimExpert Heat Exchanger Teaching Pilot Plant Pilot Plant Model SE308LW-2HS.* In this model, hot water is being generate and fed into a shell and Tube Heat Exchanger to heat up a product stream. Heated product is then cooled to a steady state by a Plate Heat Exchanger before re-entering the Product tank. Therefore it is possible to compare the performance between two types of heat exchangers. Automatic On-Off level control are applied to both the Heating Medium tank and Product Tank to ensure that each experiment may be conducted continuously.

The control system ranging from Simple On-Off level control, P-only, PI and PID control to advance Cascade Temperature control.

2.1.1 Objective

This experiment is divided into Two (2) sections:

- Section A: **Identify the Process,**
- Section B: **Open Loop Controller Tuning,**

©2005 by PCS UTM

2.1.2 Results

The student is expected to discuss a report on the findings about the process and make a comparison of the proposed tuning methods.

2.2 Preparation

STEP	ACTION	REMARKS
1	Ensure that all Utility Services are ready (i.e. Switch on Power Supply to Control panel and Switch on Air Supply System to the Pilot Plant).	
2	Open hand valve HV100. Fill the Vessel VE110 with water until it reaches the high level.	
3	Ensure that the DCS is Ready (i.e. It is communicating properly with the control panel)	
4	When the computer starts up, go in the Boot Page and then perform the login operation.	The Overview Page will appear.
5	Click on the '**Experiment 3C**' button to access Experiment 3C.	The Experiment 3C Page will appear.
6	Switch on heater HE110 and HE 111	
7	Open the Control Valve **FY131** Manually (30% open)	
8	Open or Close hand valves at the Pilot Plant as Follows: -Open Hand Valve HV151, HV131, HV132, HV141, HV111, HV125 -Close others	Hand Valves to be Open/Closed Fully
9	Start the pump P112 and P152	Hand Valves to be Open/Closed Fully
10	Stop the pump P112 and P152. When temperature of water in VE110 reach 50 degree Celsius.	

2.3 Section A: Process Identification

2.3.1 Start Experiment

STEP	ACTION	REMARKS
1	Continue the experiment from the Section above	
2	At the **TIC137 Controller Faceplate**, manually set its MV to 30%.	
3	Start Pump P112 and P152 Start CL140	
4	Click '**Exp. Config**' to start the experiment.	
5	Observe the **Temperature** curve **(TIC137)** from the Trend Window and wait until it has stabilized.	
6	At the Controller Faceplate (TIC137), manually **increase** its MV by an additional 10%	Set MV to 40 and press Enter
7	Observe the Temperature curve (TIC137) from the Trend Window and wait until it has stabilized to a new constant value. Freeze the Trend window.	This is the process Reaction curve
8	Print out the Trend curve.	Print in colour
9	Stop Pump P112 and P152 Stop CL140	
10	Click '**STOP**' to stop the experiment.	

2.4 Open Loop Controller Tuning

2.4.1 Results Analysis

1	Using the printed graph obtained from Section above, measure and tabulates the relevant values as required. Refer to **Appendix B, Table B-1**	Note dB_u and dM are change from the 1st stable output to the 2nd.
2	Base on the equations for Open Loop Tuning, calculate the required controller tuning parameters for TIC137 Refer to **Appendix C, Table C-1**	
3	At the TIC137 controller faceplates, key in the calculated controller tuning parameters.	
4	You are now ready to test the performance of the Control Loop. Proceed to the next section.	

2.5 Appendices

2.5.1 Appendix A

Process Block Diagram

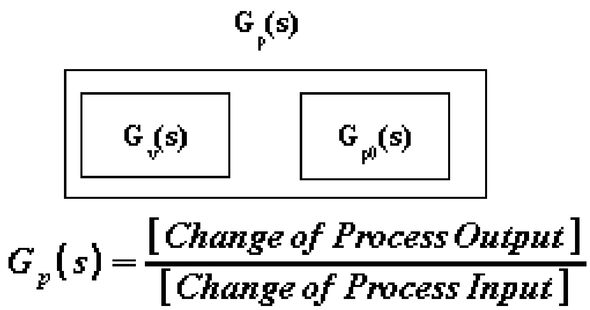


Fig A-1: Process Model inclusive of Control Valve

Process Reaction Curve 1

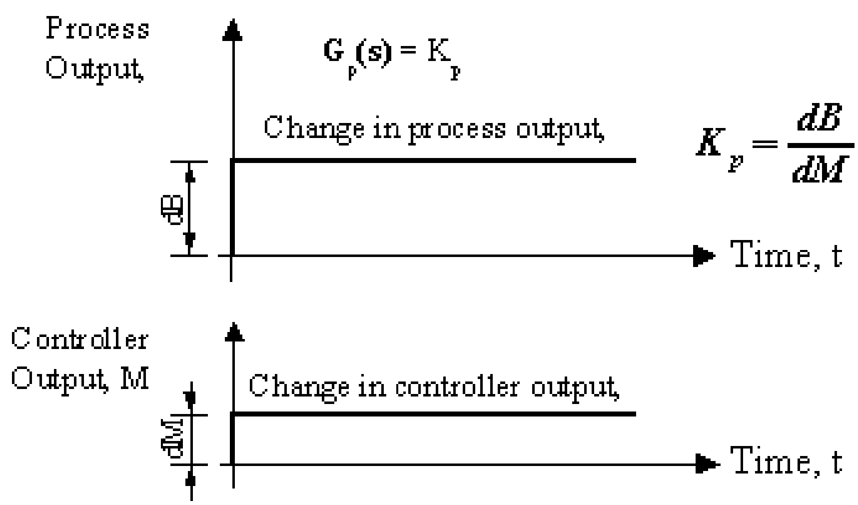


Fig A-2: Resistive Process Reaction Curve

Process Reaction Curve 2

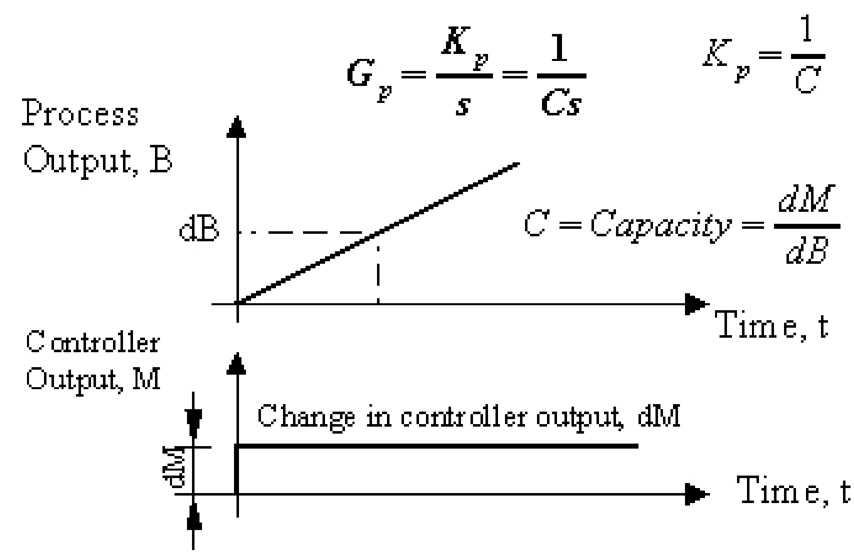


Fig A-3: Integratnig Process Reaction Curve

Process Reaction Curve 3

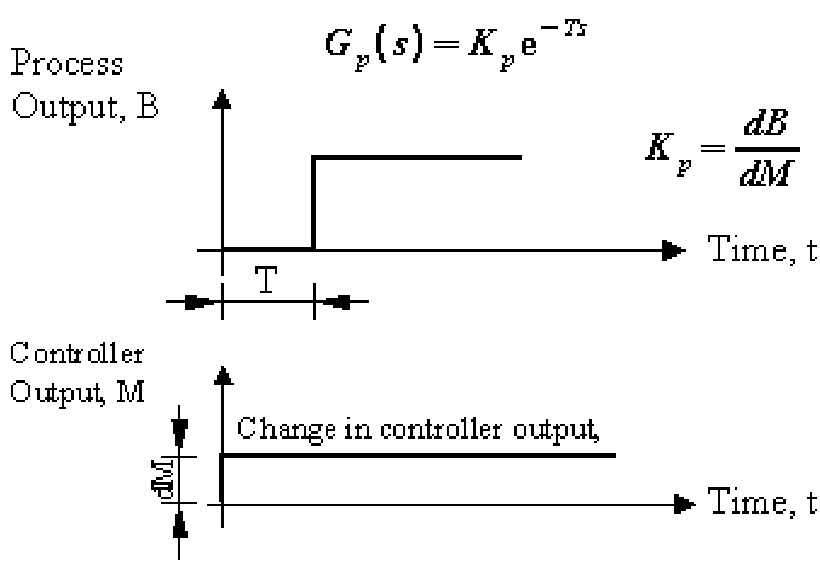


Fig A-4: Dead-time Process Reaction Curve

©2005 by PCS UTM

Process Reaction Curve 4

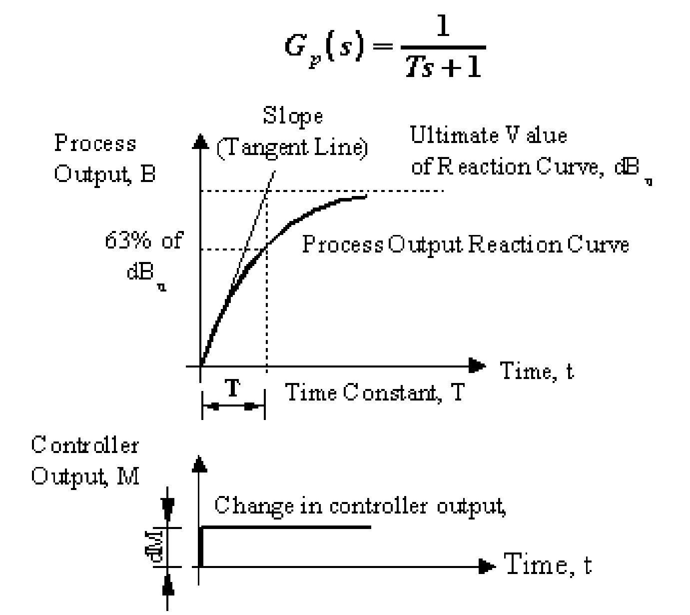


Fig A-5: First Order Process Reaction Curve

Process Reaction Curve 5

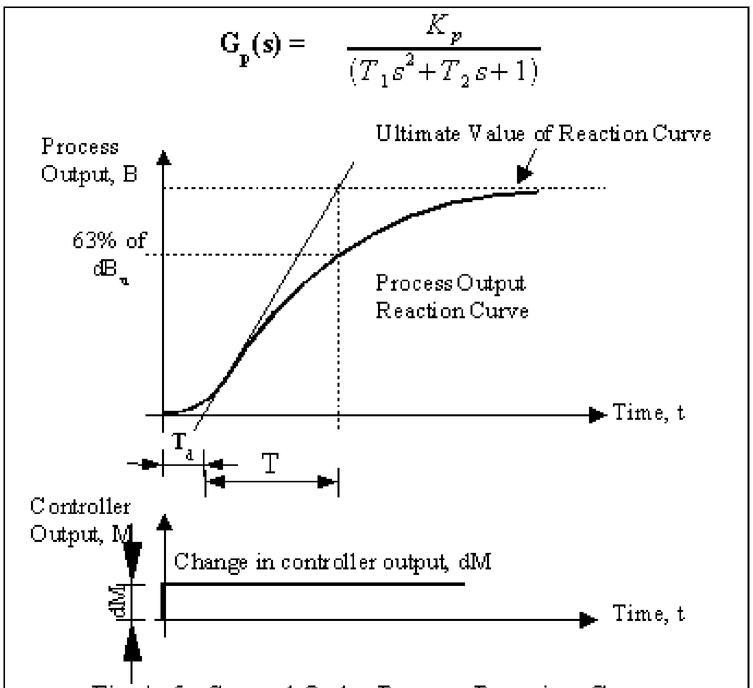


Fig A-6: Second Order Process Reaction Curve

©2005 by PCS UTM

2.5.2 Appendix B

Table B.1: Results for Open Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
Change in Manipulated Variable, dM				
Change in Ultimate Value, dB_u				
Slope S				
Apparent Dead Time, T_d				
Calculations:				
Apparent Time Constant T = dB_u/S				
Steady State Process Gain $K_P = dB_u/dM$				
$R = T_d/T$				
Tuning Parameters:				
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.2: Results for Closed Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
$K_{u,}$ Ultimate Controller Gain				
Time for 3 Oscillation periods, or more				
CALCULATIONS:				
Ultimate Time T_{u} (Time for one Oscillation period)				
Tuning Parameters:	**Test 1**	**Test 2**	**Test 3**	**Average**
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.3a: Results for Flow Compensation Calculation FZ210

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT210 , m^3/h						
Corresponding Line Pressure PT210, barg						
Corresponding Line Temperature TT210, degC						
Compensated Flow from DCS, FZ210, scmh						
Constants: q_{Fnom} , scmh	32	32	32	32	32	32
Constants: p_{fnom} , barg	5	5	5	5	5	5
Constants: t_{fnom} , degC	30	30	30	30	30	30
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_F , scmh						
% Error $\frac{FZ210 - q_F}{FZ210} x\, 100$, %						

Table B.3b: Results for Flow Compensation Calculation FZ211

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT211 , m^3/h						
Corresponding Line Pressure PT211, barg						
Corresponding Line Temperature TT211, degC						
Compensated Flow from DCS, FZ211, scmh						
Constants: q[illegible] , scmh	32	32	32	32	32	32
Constants: p[illegible] , barg	5	5	5	5	5	5
Constants: t[illegible] , degC	120	120	120	120	120	120
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_F , scmh						
% Error $\frac{FZ210 - q_F}{FZ210} x 100$						
Flow comparison, % Error $\frac{FZ211 - FZ210}{FZ210} x 100$						

©2005 by PCS UTM

Table B.5: Results for Control Valve Characterisation

Measurement	Pont 1	Pont 2	Pont 3	Pont 4	Pont 5	Average
Opening of FCV211 (%)						
Source Pressure, PT202 (bar)						
Inlet Pressure, PT211 (bar)						
Outlet Pressure, PT212 (bar)						
Exit Pressure, p_e (bar)	0	0	0	0	0	
Compensated Flow FZ211 (scmh)						
Characteristic Constant, a						
CALCULATIONS:						
Pressure across valve when it is 100% open $dp_{100} = PT211 - PT212$						
$K = q_{100}/dp_{100}$						
Total System Pressure head $dp_{sys} = PT202 - p_e$						
Max. Pressure across valve $dp_{max} = PT211 - PT212$ at minimum flow ie. When valve is 10% open, (bar)						
Min. Pressure across valve $dp_{min} = PT211 - PT212$ at maximum flow ie. When valve is 100% open, (bar)						
Relative Pressure drop across valve $R_v = dp_{min}/dp_{max}$						

Note: q_{100} is the volumetric flow at standard conditions when control valve is 100% open, in scmh

dp_{100} is the differential pressure across the control valve when it is 100% open, in bar

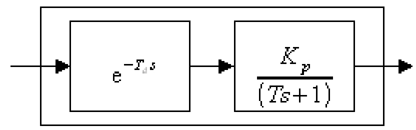
©2005 by PCS UTM

2.5.3 Appendix C

Calculation of Open Loop Tuning Parameters

Cohen-Coon Rules

The Cohen & Coon tuning rule assumes that the S-shaped Process Reaction curve can be approximated by a process model consisting of a First order lag and a Dead Time.



$$G_p(s) = \frac{[K_p e^{-T_d s}]}{(Ts+1)}$$

Fig C.1 Approximation to S-shaped Process Reaction Curve

Table C.1: Open Loop Tuning – Controller Parameter Calculations

Control Modes	**Calculation** (**R = T_L/T**)
P only	$[\frac{1}{R.Kp}][1+\frac{R}{3}]$ $K_c =$
P + I	$[\frac{1}{R.Kp}][\frac{9}{10}+\frac{R}{12}]$ $K_c =$
	$Td.\frac{(32+6R)}{(13+8R)}$ $T_I =$
P + I + D	$[\frac{1}{R.Kp}][\frac{4}{3}+\frac{R}{4}]$ $K_c =$
	$Td.\frac{(32+6R)}{(13+8R)}$ $T_I =$
	$Td.\frac{4}{(11+2R)}$ $T_D =$

©2005 by PCS UTM

Calculation of Closed Loop Tuning Parameters

Ziegler-Nichols Rules

Table C.2: Closed Loop Tuning – Controller Parameter Calculations

Control Modes	**Kc**
P only	$K_c = 0.5K_u$
P + I	$K_c = 0.45K_u$
	$T_I = T_u/1.2$
P + I + D	$K_c = 0.6K_u$
	$T_I = T_u/2$
	$T_D = T_u/8$

©2005 by PCS UTM

3

Heat Exchanger Temperature Control Pilot Plant

3.1 Heat Exchanger Temperature Control (TIC137)

Heat Exchanger-Temperature Control Pilot Plant is a control system via computer to monitor the heat exchange between hot and cool liquid inside each tank and vessels. Each process can be adjust or control using the computer either by user or itself. Its applications are widely being utilized in plant control and automation throughout the world.

The basic mechanism that is used in real industry is demonstrated in the *SimExpert Heat Exchanger Teaching Pilot Plant Pilot Plant Model SE308LW-2HS.* In this model, hot water is being generate and fed into a shell and Tube Heat Exchanger to heat up a product stream. Heated product is then cooled to a steady state by a Plate Heat Exchanger before re-entering the Product tank. Therefore it is possible to compare the performance between two types of heat exchangers. Automatic On-Off level control are applied to both the Heating Medium tank and Product Tank to ensure that each experiment may be conducted continuously.

The control system ranging from Simple On-Off level control, P-only, PI and PID control to advance Cascade Temperature control.

©2005 by PCS UTM

3.1.1 Objective

This experiment is divided into Two (2) sections:

- Section A: **Closed Loop Tuning of the simple PID Loop,**
- Section B: **Performance of Simple PID Loop.**

3.1.2 Results

The student is expected to discuss a report on the findings about the process and make a comparison of the two proposed tuning methods.

3.2 Section A: Closed Loop

3.2.1 Start Experiment: Closed Loop Tuning

STEP	ACTION	REMARKS
1	Continue the experiment from the Section above.	
2	At the **TIC137 Controller Faceplate**, manually set its MV to 30%.	
3	Start Pump P112 and P152. Start CL140	
4	Click '**Exp. Config**' to start the experiment.	
5	Observe the **PV** curve **(TT137)** from the Trend Window and wait until it has stabilized.	
6	Set **TIC137** controller to **Auto-mode.** See that it is able to control its Temperature (TT137) properly. Wait for TT137 to become stable.	
7	Set the Controller Faceplate (TT137) Gain to 32, the Integral time to 9999, and the Derivative time to 0.	
8	At the Controller Faceplate (TT137), adjust the Set Point (SP) to 45 Deg C	Adjust through MV
9	Slowly increase the controller MV to bring the Process Value (PV) to almost equal to the Set Point.	
10	Observe its Process Value (PV) from the Trend Window and wait until it has stabilized to a constant value.	
11	Set the Controller to Auto mode	
12	Wait for its Process Value (PV) to stabilize.	
13	Make a Small Step Change to the Set Point (i.e. increase the set point by 15%)	

©2005 by PCS UTM

14	Observe its Process Value (PV) from the Trend Window. If the PV response is not oscillatory, double the controller gain value until it becomes oscillatory.	
15	If the PV response is oscillatory, observe whether the magnitude of PV is increasing or decreasing. If it is increasing, reduce the controller gain by 1.5 times. If the PV is decreasing, increase the controller gain by 1.5 times. Aim to obtain an oscillatory response with almost constant amplitude.	
16	When Constant Amplitude Oscillation is achieved, allow up to 3 or more oscillation cycles to be recorded and Freeze the trend window	
17	Print out the PV response curve	Print in colour
18	Start Pump P121 and P152. Start CL140	
19	Click '**STOP**' to stop the experiment.	

3.2.2 Results Analysis

1	Using the printed graph obtained from Section above, measure and tabulates the relevant values as required. Refer **to Appendix B, Table B-2**.	
2	Base on the equations for Closed Loop Tuning, calculate the required controller tuning parameters. Refer to **Appendix C, Table C-2.**	
3	At the controller faceplate (TIC 137), key in the calculated the controller tuning parameters.	
4	You are now ready to test the performance of the Cascade Control Loop. Proceed to the next section.	

3.3 Section B: Control Loop Performance Test

3.3.1 Start Experiment

STEP	**ACTION**	**REMARKS**
1	Continue the experiment from the Section above.	
2	At the Controller Faceplate (TIC137), set it to Manual Mode. Ensure that the Controller P, I, D parameters have been set.	
3	Start the Pump P121 and P152. Start CL140	
4	Click '**Exp. Config**' to start the experiment.	
5	Adjust the Set Point (SP) of the Controller to 45 Deg C	

©2005 by PCS UTM

6	Adjust the Controller output (MV) until its Process Value (PV) is close to its Set Point (SP).	
7	Set the Controller to 'Auto' mode	
8	Observe its PV curve (TT137) from the Trend Window and wait until it is reasonable stable.	
9	Change the Set Point (SP) of the Primary Controller TIC137 by 15%.	
10	Observe its PV curve (TT137) from the Trend Window and look for some typical response characteristics. Refer to **Appendix A** for guidelines.	
11	Capture the important process response and Print out the Trend Curve	Print in colour
12	Stop the Pump, P121 and P152. Stop CL140	
13	Click '**STOP**' to stop the experiment.	

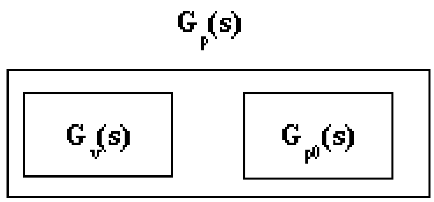
3.3.2 Tabulation and Results Analysis

1	Using the printed graph obtained from Section above, measure and tabulates the relevant as required. Refer to **Appendix B, Table B-3**.	
2	Describe the Characteristic of the Process response.	
3	Discuss the function of each controller tuning parameters, P, I, D.	
4	Suggest any improvements to the process control loop and its total error.	

3.4 Appendices

3.4.1 Appendix A

Process Block Diagram



$$G_p(s) = \frac{[\text{Change of Process Output}]}{[\text{Change of Process Input}]}$$

Fig A-1: Process Model inclusive of Control Valve

Process Reaction Curve 1

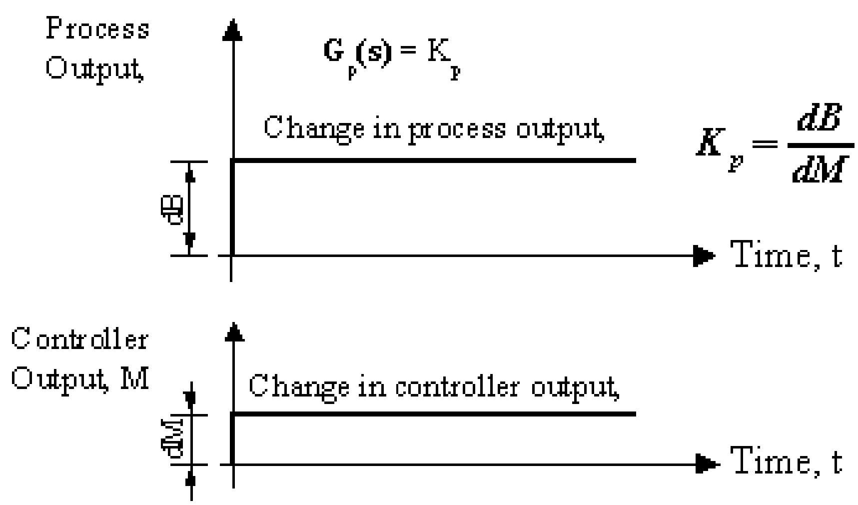


Fig A-2: Resistive Process Reaction Curve

©2005 by PCS UTM

Process Reaction Curve 2

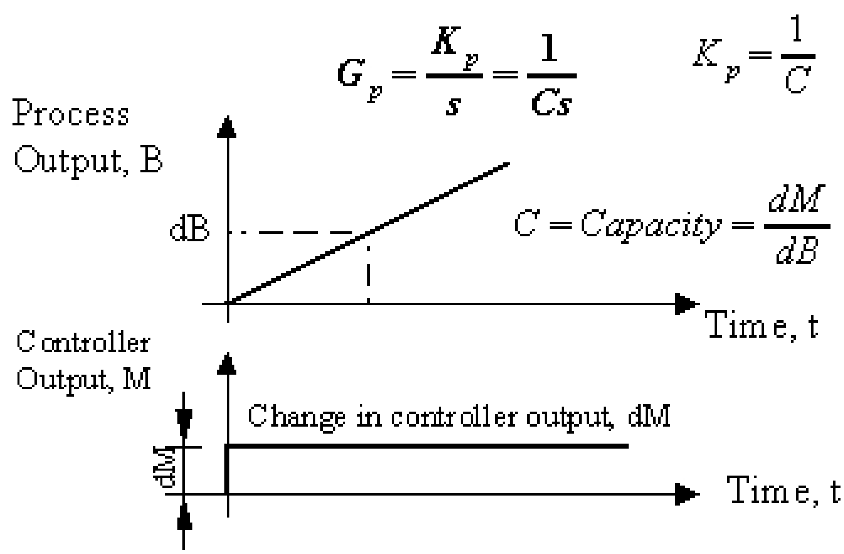


Fig A-3: Integratnig Process Reaction Curve

Process Reaction Curve 3

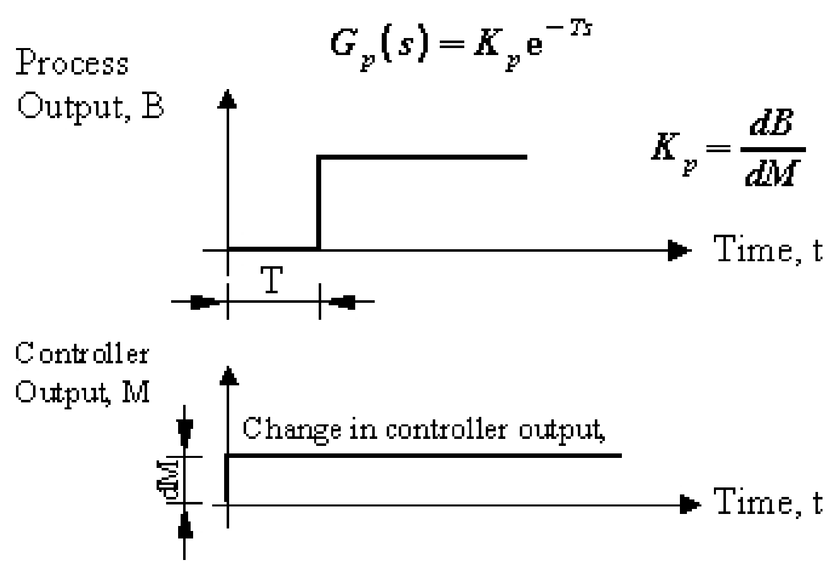


Fig A-4: Dead-time Process Reaction Curve

Process Reaction Curve 4

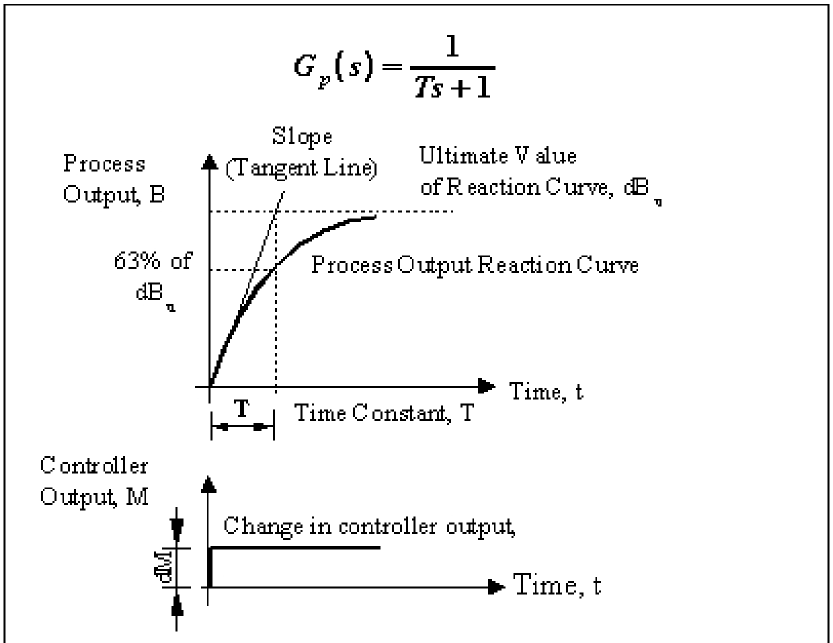


Fig A-5: First Order Process Reaction Curve

Process Reaction Curve 5

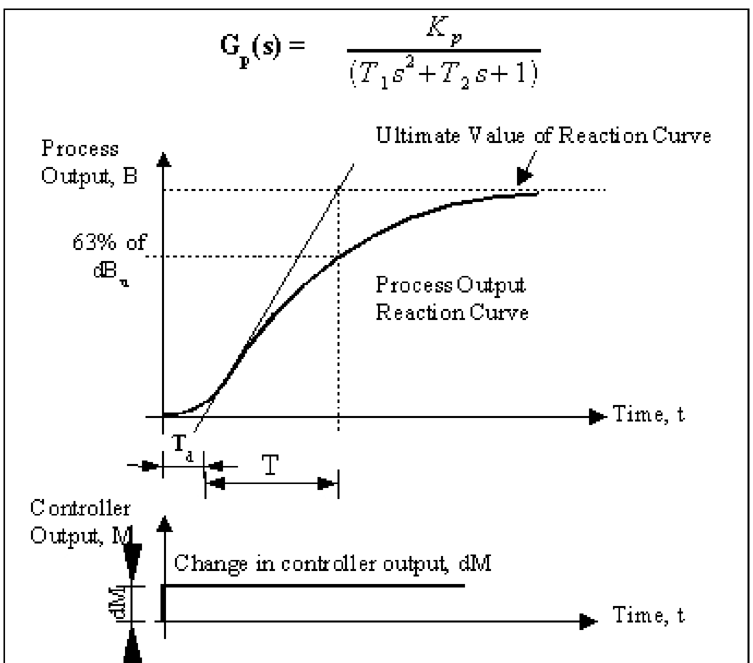


Fig A-6: Second Order Process Reaction Curve

3.4.2 Appendix B

Table B.1: Results for Open Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
Change in Manipulated Variable, dM				
Change in Ultimate Value, dB_u				
Slope S				
Apparent Dead Time, T_d				
Calculations:				
Apparent Time Constant T = dB_u/S				
Steady State Process Gain $K_P = dB_u/dM$				
$R = T_d /T$				
Tuning Parameters:				
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.2: Results for Closed Loop Tuning

Measurement	Test 1	Test 2	Test 3	Average
$K_{u,}$ Ultimate Controller Gain				
Time for 3 Oscillation periods, or more				
CALCULATIONS:				
Ultimate Time T_{u} (Time for one Oscillation period)				
Tuning Parameters:	**Test 1**	**Test 2**	**Test 3**	**Average**
K_c (Gain)				
Integral Time, T_I (minutes/repeat)				
Derivative Time, T_D (minutes/repeat)				

©2005 by PCS UTM

Table B.3a: Results for Flow Compensation Calculation FZ210

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT210 , m^3/h						
Corresponding Line Pressure PT210, barg						
Corresponding Line Temperature TT210, degC						
Compensated Flow from DCS, FZ210, scmh						
Constants: q_{Rbase} , scmh	32	32	32	32	32	32
Constants: p_{fbase} , barg	5	5	5	5	5	5
Constants: t_{fbase} , degC	30	30	30	30	30	30
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_F , scmh						
% Error $\frac{FZ210 - q_F}{FZ210} x\,100$, %						

©2005 by PCS UTM

Table B.3b: Results for Flow Compensation Calculation FZ211

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Openning of FCV211						
Uncompensated Flow FT211 , m^3/h						
Corresponding Line Pressure PT211, barg						
Corresponding Line Temperature TT211, degC						
Compensated Flow from DCS, FZ211, scmh						
Constants: q_{Fnom} , scmh	32	32	32	32	32	32
Constants: p_{fnom} , barg	5	5	5	5	5	5
Constants: t_{fnom} , degC	120	120	120	120	120	120
CALCULATIONS:						
Absolute line Pressure, P (bar)						
Absolute line Temperature, T (K)						
Compensated flow, q_F , scmh						
% Error $\frac{FZ210 - q_F}{FZ210} x\,100$						
Flow comparison, % Error $\frac{FZ211 - FZ210}{FZ210} x\,100$						

Table B.5: Results for Control Valve Characterisation

Measurement	Pont 1	Pont 2	Pont 3	Pont 4	Pont 5	Average
Opening of FCV211 (%)						
Source Pressure, PT202 (bar)						
Inlet Pressure, PT211 (bar)						
Outlet Pressure, PT212 (bar)						
Exit Pressure, p_e (bar)	0	0	0	0	0	
Compensated Flow FZ211 (scmh)						
Characteristic Constant, a						
CALCULATIONS:						
Pressure across valve when it is 100% open $dp_{100} = PT211 - PT212$						
$K = q_{100}/dp_{100}$						
Total System Pressure head $dp_{sys} = PT202 - p_e$						
Max. Pressure across valve $dp_{max} = PT211 - PT212$ at minimum flow ie. When valve is 10% open, (bar)						
Min. Pressure across valve $dp_{min} = PT211 - PT212$ at maximum flow ie. When valve is 100% open, (bar)						
Relative Pressure drop across valve $R_v = dp_{max}/dp_{min}$						

Note: q_{100} is the volumetric flow at standard conditions when control valve is 100% open, in scmh

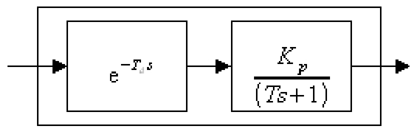
dp_{100} is the differential pressure across the control valve when it is 100% open, in bar

3.4.3 Appendix C

Calculation of Open Loop Tuning Parameters

Cohen-Coon Rules

The Cohen & Coon tuning rule assumes that the S-shaped Process Reaction curve can be approximated by a process model consisting of a First order lag and a Dead Time.



$$G_p(s) = \frac{[K_p e^{-T_d s}]}{(Ts+1)}$$

Fig C.1 Approximation to S-shaped Process Reaction Curve

Table C.1: Open Loop Tuning – Controller Parameter Calculations

Control Modes	**Calculation**	**(R = T_L/T)**
P only	$[\frac{1}{R.Kp}][1+\frac{R}{3}]$	K_c =
P + I	$[\frac{1}{R.Kp}][\frac{9}{10}+\frac{R}{12}]$	K_c =
	$Td.\frac{(32+6R)}{(13+8R)}$	T_I =
P + I + D	$[\frac{1}{R.Kp}][\frac{4}{3}+\frac{R}{4}]$	K_c =
	$Td.\frac{(32+6R)}{(13+8R)}$	T_I =
	$Td.\frac{4}{(11+2R)}$	T_D =

©2005 by PCS UTM

Calculation of Closed Loop Tuning Parameters

Ziegler-Nichols Rules

Table C.2: Closed Loop Tuning – Controller Parameter Calculations

Control Modes	**Kc**
P only	$K_c = 0.5K_u$
P + I	$K_c = 0.45K_u$
	$T_{I\,=}\,T_u/1.2$
P + I + D	$K_c = 0.6K_u$
	$T_I = T_u/2$
	$T_D = T_u/8$

4

Flow Control Pilot Plant

4.1 Objective

This module gives some instruction how to use the FLOW CONTROL PROCESS developed using GENESIS 32. In this module, transient response data will be collected which will be used to obtain a first-order-plus-time-delay transfer function model of the process. This model will later serve as the basis for several types of feedback controller design. The performance of a feedback control system depends on the values of the controller tuning constants. The Proportional-Integral-Derivative (PID) controller will be used in this module. A trial and error selection process for PID controller tuning constants requires a lengthy iterative procedure. In this module, you will be introduced to standard algorithms which produce good initial estimates of K_c, τ_I, and τ_D.

4.2 Main

1. Double click **Launcher.gdf** to launch the flow control software. The INTRO screen will be appeared.

©2005 by PCS UTM



Figure 4.1: Intro Screen

2. Press [HOME] or click the button to enter to Main Menu.

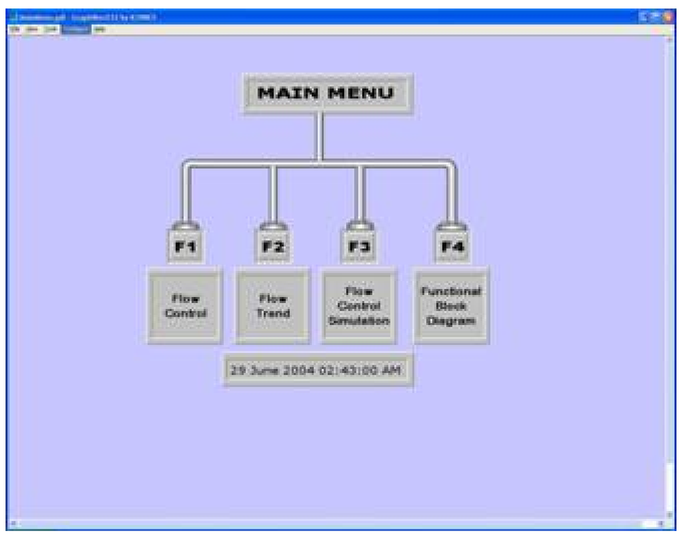


Figure 4.2: Main Menu Screen

3. The main menu screen will display 4 selections. Each of can be selected either clicking on its button or press the key (F1 -> F4)

   · **[F1] Flow Control**

   This feature allows the user to start the genesis of flow control process. The genesis will interact with the actual flow control machine through the serial communication.

- **[F2] Flow Trend**

  This feature is the additional part where output of the flow control process from the machine will be generated and analyses into graph and database.

- **[F3] Flow Control Simulation**

  This feature is the same as **'Flow Control'** but the data for the flow control process is synthetically generated by the computer as a simulation.

- **[F4] Functional Block Diagram**

  This feature will show rough picture of hardware and its functionality

4.3 Display

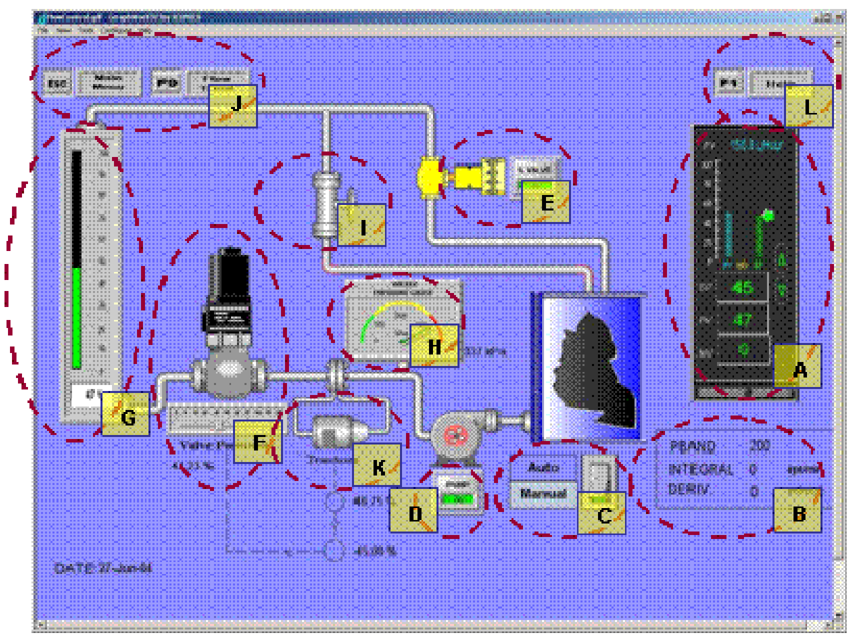


Figure 4.3: Flow Control Screen

A (Control Panel)

SP (Set Point)	PID set point. Only work in Auto Mode
PV (Process Variable)	The actual flow rate, read from flow transducer
MV (Manipulated Variable)	Positional valve value. Only work in Manual Mode

** All values displayed in percentage (%)*

B (PID Constants)

- All PID constants in x100 scale

C (PID Switch)

-

D (Pump Switch)

-

E (Solenoid Valve Switch)

-

F (Positional Valve Indicator)

- Value in percentage (%)

G (Water Flow Indicator)

- Value in percentage (%)

H (Meter Pressure Indicator)

- Give current pressure in kilo Pascal

I (Manual Valve)

- Can't be controlled by software in Genesis 32.

J (Menus)

K (Flow Transducer)

- Give feedback to hardware as current flow rate value.

L (Help)

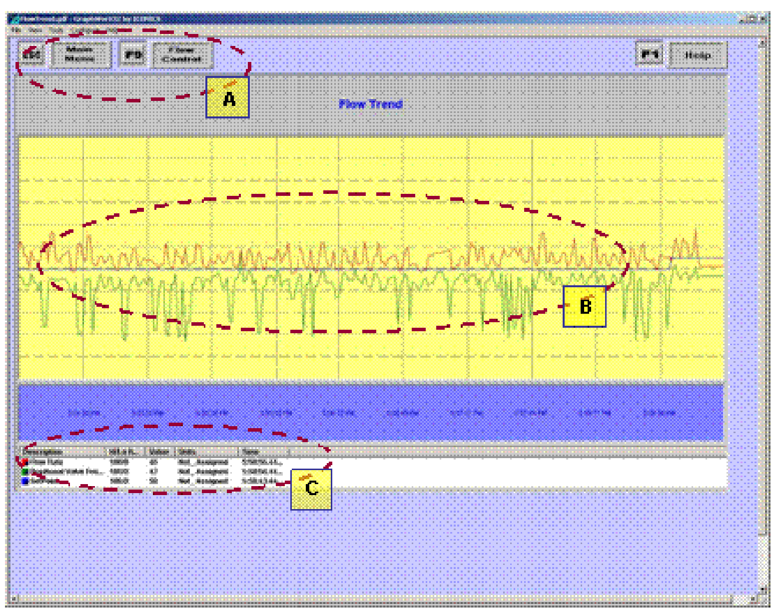


Figure 4.4: Flow Trend Screen

A (Menus)

B (Graph Canvas)
Plot 3 types of data in percentage (%): Set Point, Flow rate and Positional Valve Value

C (Indicator)
- For reference

This trend menu will appeared when the graph (B area) was double clicked.

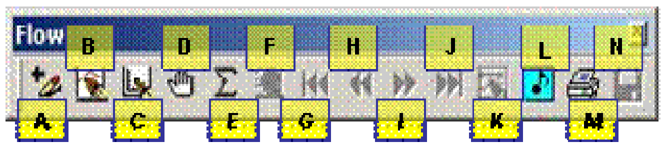


Figure 4.5: Trend Menu

A – Edit pens	H – Cursor back
B – Edit trend	I – Cursor forward
C – Edit period	J – Page forward
D – Freeze mode	K – Set right time
E – Statistics	L – Show comments
F – Zoom trend	M – Print trend
G – Page back	N – Save trend

** Please refer to Iconic Help for detail explanations*

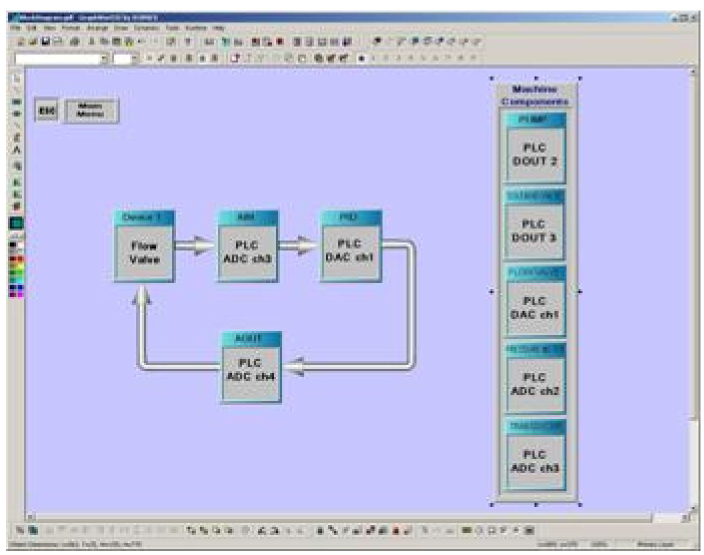


Figure 4.6: Block Diagram Screen

Description
- Just show the block diagram of the system

4.4 Step By Step Operation

In this section, the reader will be guided step by step to do the flow rate experiment.

4.4.1 Startup

1. Switch ON power to all equipments including computer
2. Double click **Launcher.gdf**
3. When **INTRO SCREEN** appeared, press [HOME] or click the button. The **MAIN MENU** screen will be appeared
4. Select the function key [F1] to [F4] to go to different pages or just click on the button

4.4.2 Flow Control

1. By selecting [F1] or **FLOW CONTROL**, page will be displayed.
2. When the indicator value such as pressure meter valid (show the number, not asterisk (*)), press GREEN button beside controller box (make sure the light turn to off).
3. Use [Tab] or [Shift + Tab] to move the cursor or just click using mouse.
4. To enter the value, type it in the cursor and press [Enter] or by dragging the scroll
5. Before starting the experiment, turn on the data logger to enable graph plotting (FLOW TREND). To turn on the data logger, follow this :
   a. Click Start -> Iconics Genesis-32 -> TrendWorX32 -> TrendWorX32 Configurator



Figure 4.7: TrendWorX32 Configurator Link

   b. Click **LogDatabase**

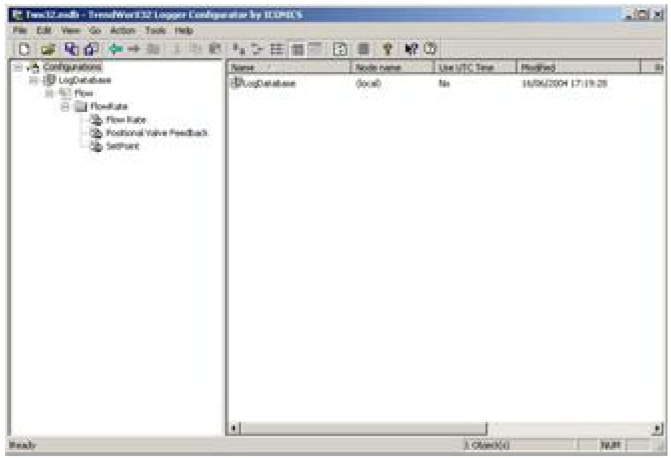


Figure 4.8: TrendWorX32 Configurator Screen

c. Press **Start Datalogging** to start the data logger

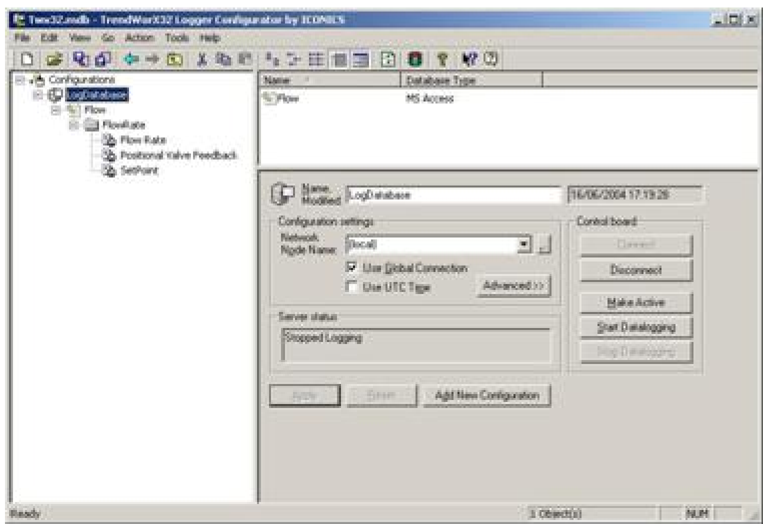


Figure 4.9: TrendWorX32 Configurator Screen 2

d. If you want to change the database name or location, click **FLOW** tree and select **BROWSE.**

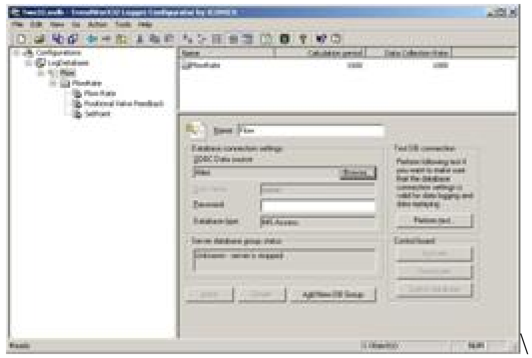


Figure 4.10: Configure Database

6. To run or stop the pump, move the cursor to that ON/OFF button and then press [Enter] or just click at that button.
7. To run or stop the solenoid valve, move the cursor to that ON/OFF button and then press [Enter] or just click at that button.
8. To change the system mode to AUTO or MANUAL, move the cursor to AUTO/MANUAL button and then press [Enter] or just click at that button.
9. To change the positional valve output manually, make sure the system run in **MANUAL MODE**. Move cursor to MV (Manipulated Value), enter the value and then press [Enter].
10. To enable PID control, make sure the system run in **AUTO MODE**. Move the cursor to PID constants box to change the PID constants value. Type the value and press [Enter].

©2005 by PCS UTM

5

Steady State Feedback Control

5.1 Introduction to the Virtual Process Control Lab (VPCL)

5.1.1 Objective

The purpose of this module is to introduce MATLAB and SIMULINK, as well as explain the basic operation of the Virtual Process Control Lab (VPCL).

5.1.2 What Is MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualisation
- Scientific and engineering graphics
- Application development, including graphical user interface building

©2005 by PCS UTM

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a programming a scalar noninteractive language such as C or Fortran.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialised technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

5.1.3 Simulink

Simulink is a software package for modelling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modelled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.

For modelling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With the interface, you can draw the models just as you would with pencil and paper (or as most textbooks depict them). This is a far cry from the previous simulation packages that require you to formulate differential equations and difference equations in a language or program. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. You can also customise and create your own blocks.

Models are hierarchical, so you can build models using top-down and bottom-up approaches. You can view the system at high level, then double-click blocks to go down through the levels to see increasing levels of model detail. This approach provides insight into how a model is organised and how its parts interact.

After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations. Using scopes and other display blocks, you can see the simulation results while the simulation is running. In addition, you can change parameters and immediately see what happen. Because MATLAB and Simulink are integrated, you can simulate, analyse, and revise your models in either environment at any point.

©2005 by PCS UTM

5.1.4 Overview of VPCL

The units of VPCL are organized around the following three elements:

1. Process modelling
2. PID Controller Design and Tuning
3. Analysis and Advanced Control Design

Throughout the course, experiments involving each of these elements will be performed on the furnace or binary distillation column. Since all of these elements interact, the final design may involve several iterations.

The Virtual Process Control Lab (VPCL) use programs that were developed to be used with SIMULINK and MATLAB. To start the Modules, type *mainmenu*. The display in Figure 5.1 will appear on the screen. From this menu you can access the various programs that will be used in the Virtual Process Control Lab. In the next Modules, we will use the *Furnace* and the *Distillation Column* models to demonstrate basic principles in process behaviour. If one selects the *Furnace* button, then the furnace menu shown in Figure 5.2 will appear on the screen. Clicking on the *Distillation Column* button will cause Figure 5.3 to appear on the screen.

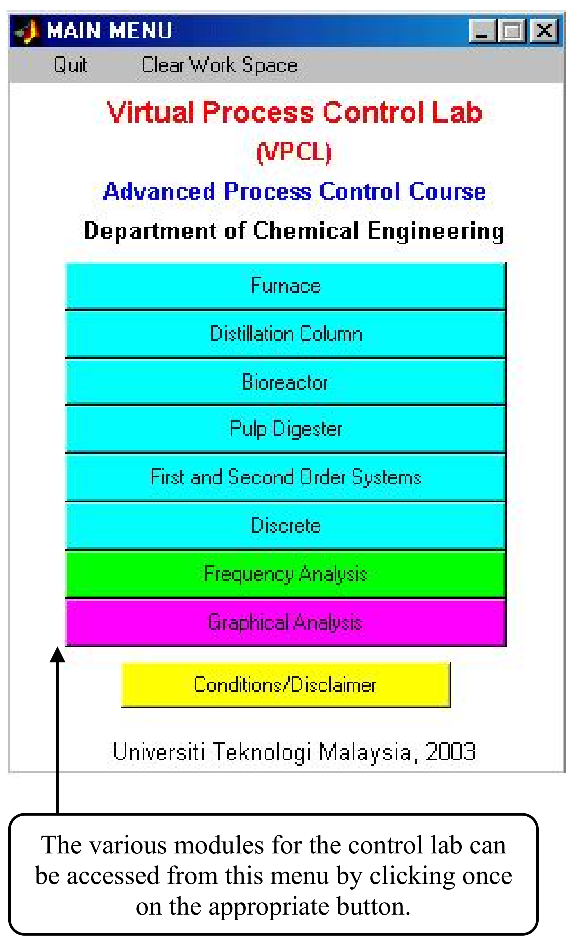


Figure 5.1: Main Menu for Virtual Process Control Lab

©2005 by PCS UTM

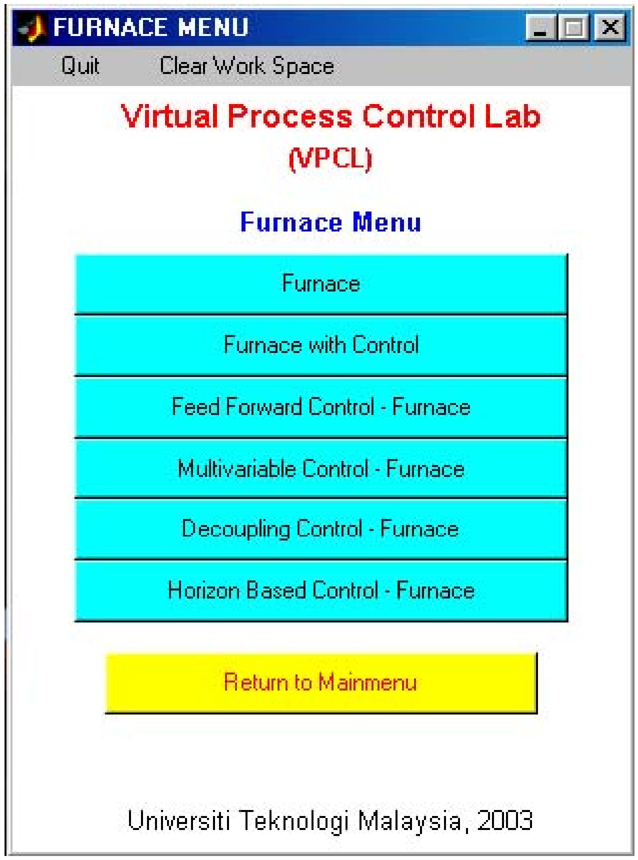


Figure 5.2: Furnace Menu

5.1.5 Ending Session

To end the session select *Yes* under the *Quit* menu from the Main Menu. This will take you back to the MATLAB prompt. At this prompt, type *quit* to exit MATLAB.

5.1.6 Summary

By the end of this Module, you should have learned how to log in at the computer. You have been introduced to MATLAB and should be able to start MATLAB and know how to run the demonstration. You should also be able to start the Virtual Process Control Lab.

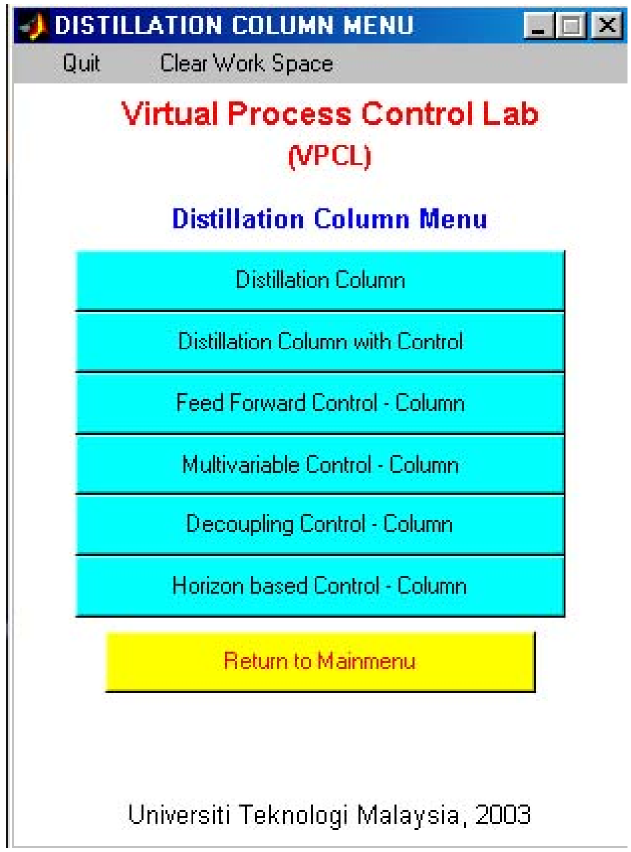


Figure 5.3: Distillation Column Menu

5.2 Objective

In this module the characteristics of some of the various types of feedback control action and their influence on the performance of the Furnace will be studied. Of particular interest is the impact of controller gain and reset time on the offset between the output and the setpoint at steady state.

5.3 Introduction

The two types of controllers that will be studied in this section are proportional (P) and proportional-integral (PI) controllers. Proportional controllers consist of a constant gain which operates on the error signal generated by subtracting the current value of the controlled variable from the setpoint. The mathematical definition of proportional control is as follows:

$$u = K_c * e$$

where:

e	is $Y_{sp} - Y_{current}$;
Y_{sp}	is the setpoint of the controlled variable;
$Y_{current}$	is the current value of the controlled variable;
u	is the manipulated variable; and
K_c	is the proportional gain

Proportional-integral controllers have the same constant gain element as proportional controllers; however, PI controllers also have an integral element. Integral action forces the controller to continue changing as long as there is an error. The mathematical definition of a proportional-integral controller is as follows:

$$u = K_c * e + \frac{K_c}{\tau_I} * \int_0^t e(t')dt'$$

where:

τ_I	is the integral or reset time constant

5.4 Procedure-Furnace

5.4.1 Proportional Control

1. To start the furnace with control, click once on the *Furnace with Control* button on the Furnace Menu (Figure 5.2). Double-click on the concentration controller block and set the controller parameters to the following values (see Figure 5.4):

Gain of the controller (K_c)	= 4.0 m^6/mol min
Reset time or the integral time constant (τ_I)	= 1 min
Rate time or the derivative time constant (τ_D)	= 0
Integral Action On	= 0
Derivative Action On	= 0

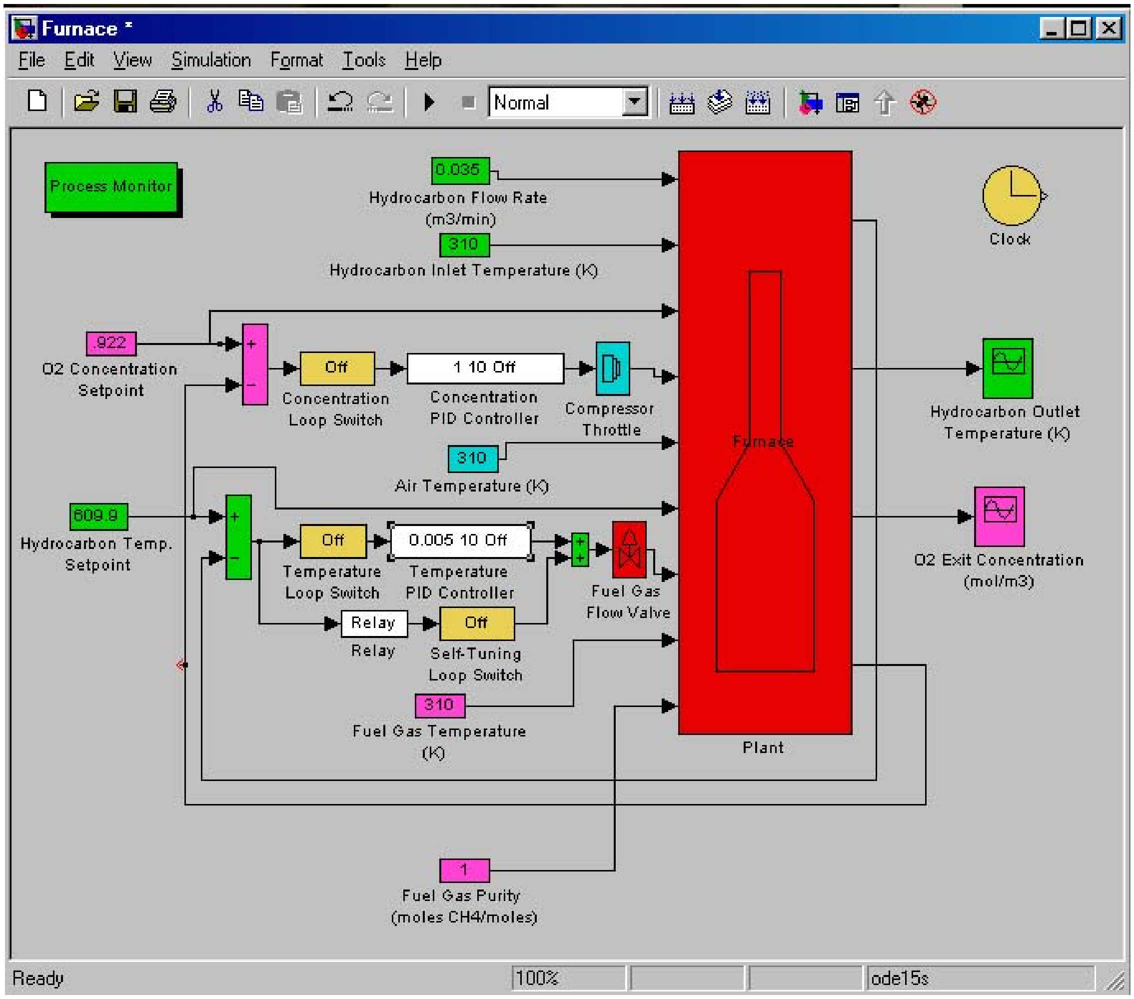


Figure 5.4: SIMULINK Flowsheet of the Furnace with Control

Exercise 1 Enter the initial values of the process variables in Table 1.

2. Double-click on the concentration loop switch and change its value to 1. Change the setpoint of *Oxygen Exit Concentration* to 1.5 mol/m^3. When the steady state value is reached, record the final values in Table 1. Note the closed-loop behaviour under proportional control.

3. From the data recorded in the table, it can be seen that the final steady state value is not equal to the setpoint. There is a steady state error which is called an offset. This is the principle drawback of the proportional controller. From the plots on the Process Monitor, one can observe that the *Hydrogen Outlet Temperature* is also affected by the manipulation of the *Air Flow Rate*. This issue will be addressed later in this module with the use of PI control.

4. Disconnect the controller by turning off the concentration loop switch. Bring the setpoint back to 0.922 mol/m^3. Allow the system to return to its initial steady state.

5. Change the controller setting to K_c = 8.45, turn on the concentration loop, and repeat the exercise from Step 2 through Step 4. Then, try K_c = 13.5 and repeat again. Be sure to record the values in Table 1 after each run.

Exercise 2 Discuss the shortcomings of proportional control observed in this exercise.

Exercise 3 What effect did increasing the proportional gain have on the performance of the proportional controller (in terms of offset or oscillatory response)?

5.4.2 Proportional Control for a Disturbance

6. Disconnect the controller by turning off the concentration loop. Bring the furnace back to its initial steady state and change the *Oxygen Exit Concentration Setpoint* back to 0.922 mol/m^3. Set the tuning parameters of the controller to the following values:

Gain of the controller (K_c)	= 4.0 m^6/mol min
Reset time or the integral time constant (τ_I)	= 1 min
Rate time or the derivative time constant (τ_D)	= 0
Integral Action On	= 0
Derivative Action On	= 0

7. Double-click on the concentration loop switch and change its value to 1. Change the *Fuel Gas Purity* from 1.0 mol CH_4/mol to 0.9 mol CH_4/mol.

Exercise 4 Record the final steady state values of the *Oxygen Exit Concentration* in Table 2.

8. Again disconnect the controller by turning off the concentration loop switch. Change the *Fuel Gas Purity* to 1.0 mol CH_4/mol.

9. Once the furnace has returned to steady state, set the gain K_c = 8.45 and turn on the concentration loop switch. Repeat the exercise from Step 7. Change the gain K_c = 13.5. Repeat Step 7 again. Be sure to enter all of the data in Table 2.

5.4.3 Proportional-Integral (PI) Control

10. Bring the system to its initial steady state condition.

11. Change the controller setting to the following values:

Gain of the controller (K_c)	= 6.45 m^6/mol min

©2005 by PCS UTM

Reset time or the integral time constant (τ_I)	=	2 min
Rate time or the derivative time constant (τ_D)	=	0
Integral Action On	=	1
Derivative Action On	=	0

12. Introduce a step change in the setpoint from 0.922mol/m^3 to 1.5 mol/m^3.

13. When steady state is reached note the final value of the *Oxygen Exit Concentration*. It should be equal to the setpoint indicating the absence of offset when the integral action is present in the controller. Also note the overshoot (if any) and the settling time of the system.

    Exercise 5 Record your data in Table 3.

14. Change the *Oxygen Exit Concentration Setpoint* back to 0.922 mol/m^3 and let the system return to its initial steady state.

15. Repeat Steps 12, 13, and 14 for the following values of the integral reset time: τ_I = 5.0, 10.0, 20.0 (min). Be sure to record your data in Table 3.3.

    Exercise 6 Describe the difference between closed-loop responses obtained for the proportional (only) versus proportional-integral controllers.

5.4.4 Proportional-Integral (PI) Control for a Disturbance

16. Bring the system to its initial steady state.

17. Change the controller settings to the following values:

Gain of the controller (K_c)	=	6.45 m^6/mol min
Reset time or the integral time constant (τ_I)	=	2 min
Rate time or the derivative time constant (τ_D)	=	0
Integral Action On	=	1
Derivative Action On	=	0

18. Keep the *Oxygen Exit Concentration Setpoint* at 0.922 mol/m^3. Introduce a step change in the *Fuel Gas Purity* from 1.0 mol CH_4/mol to 0.9 mol CH_4/mol.

19. When steady state is reached note the final value of the *Oxygen Exit Concentration*. It should be equal to the setpoint indicating the absence of offset when integral action is present in the controller. Also note the overshoot (if any) and the settling time of the system.

    Exercise 7 Record your data in Table 4

20. Change the *Fuel Gas Purity* back to 1.0 mol CH_4/mol and let the system return to its initial steady state.

21. Repeat Steps 18, 19, and 20 for the following values of the integral reset time: τ_I = 5.0, 10.0, 20.0 (min). Be sure to record your data in Table 3.4.

    Exercise 8 Describe the difference between closed-loop responses obtained for the proportional (only) versus proportional-integral controllers.

5.5 Summary

In this unit you have developed proportional and proportional-integral feedback controllers which automatically adjust the output variables by making changes in the manipulated variables. You have also seen that the feedback control system exhibits oscillatory behaviour for certain control settings. As the controller gain is increased to eliminate steady state offset, such oscillatory behaviour can become pronounced and for certain controller settings the closed-loop system can become unstable. In order to adjust or tune the controller settings, one must have information not only about the steady state behaviour of the process, but also about its dynamic characteristics.

©2005 by PCS UTM

5.6 Appendix

Section: Approved by: ______________

Group:

Date:

Exercise 1

Table 1 Oxygen Exit Concentration – P Control Servo Response

K_c m^6/mol min	Steady State Value	Setpoint Value
0 (initial)		
4		
8.45		
13.5		

Exercise 2

Exercise 3

©2005 by PCS UTM

Exercise 4

Table 2 Oxygen Exit Concentration – P Control Load Response

K_c m^6/mol min	Steady State Value	Setpoint Value
0 (initial)		
4		
8.45		
13.5		

Exercise 5

Table 3 Oxygen Exit Concentration – PI Control Servo Response

τ_I (min)	Steady State Value	Setpoint Value
2 (initial)		
5		
10		
20		

Exercise 6

Exercise 7

Table 4 Oxygen Exit Concentration – PI Control Load Response

τ_I (min)	Steady State Value	Setpoint Value
2 (initial)		
5		
10		
20		

Exercise 8

©2005 by PCS UTM

6

Distillation Controller Tuning

6.1 Introduction to the Virtual Process Control Lab (VPCL)

6.1.1 Objective

The purpose of this module is to introduce MATLAB and SIMULINK, as well as explain the basic operation of the Virtual Process Control Lab (VPCL).

6.1.2 What Is MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualisation
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a programming a scalar noninteractive language such as C or Fortran.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialised technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

6.1.3 Simulink

Simulink is a software package for modelling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modelled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.

For modelling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With the interface, you can draw the models just as you would with pencil and paper (or as most textbooks depict them). This is a far cry from the previous simulation packages that require you to formulate differential equations and difference equations in a language or program. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. You can also customise and create your own blocks.

Models are hierarchical, so you can build models using top-down and bottom-up approaches. You can view the system at high level, then double-click blocks to go down through the levels to see increasing levels of model detail. This approach provides insight into how a model is organised and how its parts interact.

After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations. Using scopes and other display blocks, you can see the simulation results while the simulation is running. In addition, you can change parameters and immediately see what happen. Because MATLAB and Simulink are integrated, you can simulate, analyse, and revise your models in either environment at any point.

©2005 by PCS UTM

6.1.4 Overview of VPCL

The units of VPCL are organized around the following three elements:

1. Process modelling
2. PID Controller Design and Tuning
3. Analysis and Advanced Control Design

Throughout the course, experiments involving each of these elements will be performed on the furnace or binary distillation column. Since all of these elements interact, the final design may involve several iterations.

The Virtual Process Control Lab (VPCL) use programs that were developed to be used with SIMULINK and MATLAB. To start the Modules, type *mainmenu*. The display in Figure 6.1 will appear on the screen. From this menu you can access the various programs that will be used in the Virtual Process Control Lab. In the next Modules, we will use the *Furnace* and the *Distillation Column* models to demonstrate basic principles in process behaviour. If one selects the *Furnace* button, then the furnace menu shown in Figure 6.2 will appear on the screen. Clicking on the *Distillation Column* button will cause Figure 6.3 to appear on the screen.

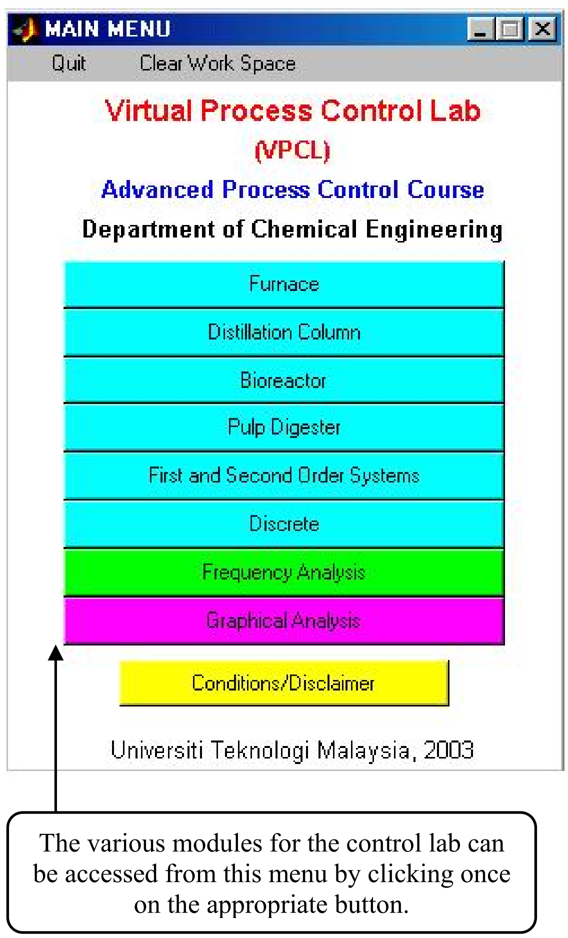


Figure 6.1: Main Menu for Virtual Process Control Lab

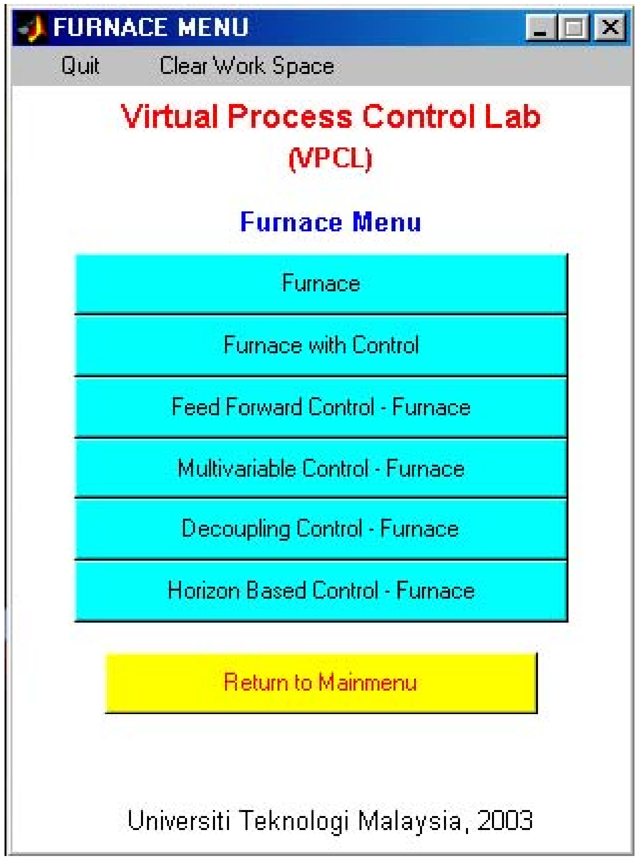


Figure 6.2: Furnace Menu

6.1.5 Ending Session

To end the session select *Yes* under the *Quit* menu from the Main Menu. This will take you back to the MATLAB prompt. At this prompt, type *quit* to exit MATLAB.

6.1.6 Summary

By the end of this Module, you should have learned how to log in at the computer. You have been introduced to MATLAB and should be able to start MATLAB and know how to run the demonstration. You should also be able to start the Virtual Process Control Lab.

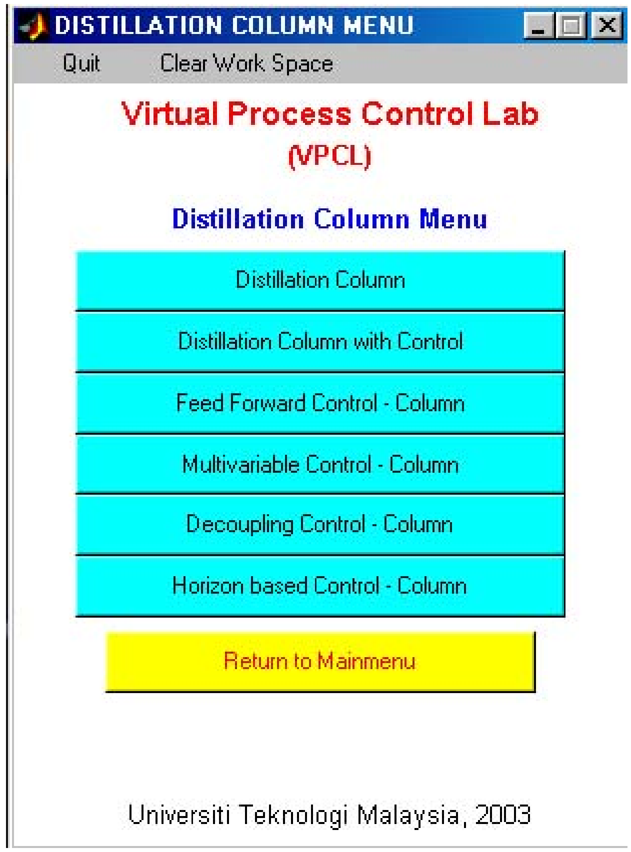


Figure 6.3: Distillation Column Menu

6.2 Objective

In this module, transient response data will be collected which will be used to obtain a first-order-plus-time-delay transfer function model of the Column. This model will later serve as the basis for several types of feedback and feedforward controller design. The performance of a feedback control system depends on the values of the controller tuning constants. The Proportional-Integral-Derivative (PID) controller will be used in this module for the Column. A trial and error selection process for PID controller tuning constants requires a lengthy iterative procedure. In this module, you will be introduced to standard algorithms which produce good initial estimates of K_c, τ_I, and τ_D.

6.3 First-Order-Plus-Time-Delay (FOTD) Transfer Function

6.3.1 Introduction

Typically, the transient response of a plant (even a nonlinear process such as the column) can be approximately modelled with a first-order-plus-time-delay (FOTD) transfer function. The mathematical description of a FOTD transfer function is as follows:

$$G_p(s) = \frac{K_p e^{-\theta s}}{\tau_p s + 1}$$

where: K_p is the system gain, θ is the time delay of the system, and τ_p is the system time constant.

Once a step change has been made in one of the inputs, the system gain, time constant. And the time delay may be calculated from the system response. The method for obtaining these parameters is described in most undergraduate chemical process dynamics and control textbooks such as 28%-63% method.

6.3.2 Procedure - Column

Data Collection

In this section, the data required for obtaining a FOTD transfer function model of the column will be collected. Open *Distillation Column* at Distillation Column Menu. Two runs will be conducted. Each run will consist of:

1. Changing one of the input variables
2. Recording the time at which the input was changed
3. Allowing the column to reach a new steady state
4. Saving the four outputs to a data file
5. Returning the input variables to their initial values
6. Allowing the column to return to its initial steady state.

The first input variable that will be changed is the *Vapour Flow Rate* (from 0.033 to 0.05 mol/sec). A detailed explanation of the data collection procedure will be given for this run.

©2005 by PCS UTM

(a) Open the clock display. Make sure to arrange the windows such that the clock display is completely visible.

(b) Choose a file name for the data from this run to be saved by typing a filename in the box located in the lower right corner of the process monitor (example: *vapourflow*), see Figure 6.4.

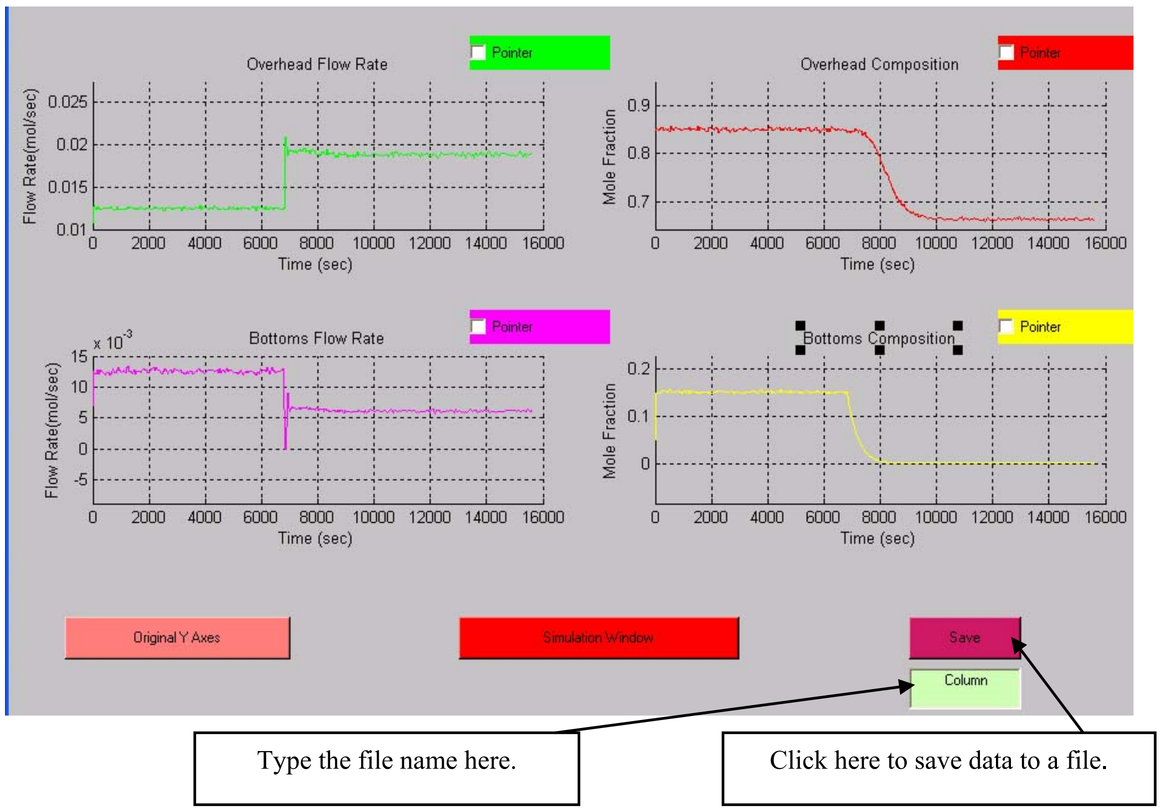


Figure 6.4: Column Process Monitor

(c) Record the filename for the current data run.

(d) Double-click on the *Vapour Flow Rate* box with the left mouse button.

(e) Double-click in the white box to highlight the value to be changed.

(f) Record the time at which the *Vapour Flow Rate* change is implemented. **Be careful: the *Vapour Flow Rate* change is not implemented until you hit *apply* or *close* with the mouse.** You should notice a slight pause in the simulation at the time in which the *Vapour Flow Rate* change is implemented.

(g) Once the column has reached steady state, click on the *Save* button located in the lower right corner of the process monitor, see Figure 6.4, to save the data to a data file. **Be careful not to wait too long because any data that is lost when the axes rescale is not recoverable.**

(h) Return the *Vapour Flow Rate* to its original value (0.033 mol/sec) and allow the column to return to its original steady state before making the next input change. **You must allow the column to return to its original steady state before making any further changes in the input variables.**

Repeat Steps (b) – (h) for the next run.
CAUTION: Choose a different filename for each run. Otherwise, the data from the previous runs will be lost.

1. Increase the value of the *Vapour Flow Rate* (from 0.033 to 0.05 mol/sec). The column should reach a new steady state in approximately 6000 seconds.

   Exercise 1
   Name of data file for this run __________________ (example: *vapourflow*).
   Time at which the *Vapour Flow Rate* change is introduced _____________ sec (from clock display).

2. Increase the value of the *Reflux Ratio* (from 1.75 to 2.5). The column should reach a new steady state in approximately 6000 simulation seconds.

   Exercise 2
   Name of data file for this run __________________ (example: *reflux*).
   Time at which the *Reflux Ratio* change is introduced _____________ sec (from clock display).

Modelling the Column

The data files collected in the previous portion of this module will be loaded into a specially designed graphical utility, called *PCMplot* to obtain transient plots of the output variables. To start *PCMplot*, return to main menu and click on the *Graphical Analysis* button in the lower right corner. A new menu, the Graphical Analysis Menu, will appear in the centre of your screen. Click on the *PCMplot – Column* button to open *PCMplot*. A graph similar to the one shown in Figure 6.5 will appear.
The procedure for loading a data file will be summarised below.

3. Click on the *Data* option on the graphics window, see Figure 6.5, and hold down the left mouse button. By moving the mouse downward, a menu with the following options will appear: *Load, Print,* and *Quit*. To load a file, continue holding the left mouse button and pull the mouse down to highlight the *Load* option. Release the mouse button.

   (a) A window title *Load Data File* should appear. Click on the data file you wish to load.

   (b) If the file loaded successfully, the following text will be displayed in your MATLAB window:
   Load selected filename

©2005 by PCS UTM

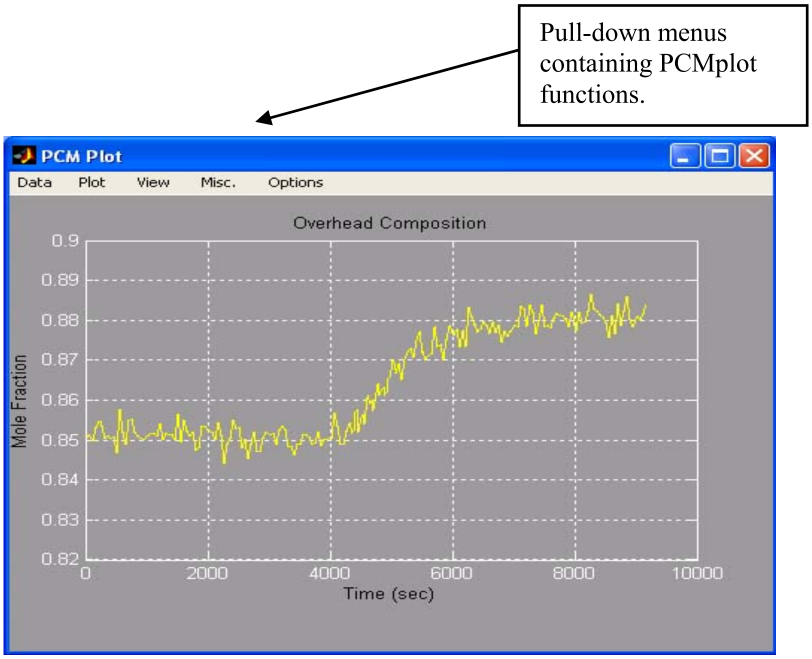


Figure 6.5: PCMplot Graphics Window

(c) Once the data has been loaded, one can plot one of the output variables by clicking on the *Plot* option at the top of the graphics window with your left mouse button. Hold the left mouse button down and pull the mouse down to obtain the following options: (i) Plot Var 1, (ii) Plot Var 2, (iii) Plot Var 3, and (iv) Plot Var 4. These options are responsible for plotting the four outputs as follows:

i. Plot Var 1 = Plot *Overhead Flow Rate*
ii. Plot Var 2 = Plot *Overhead MeOH Composition*
iii. Plot Var 3 = Plot *Bottom Flow Rate*
iv. Plot Var 4 = Plot *Bottom MeOH Composition*

(d) To plot the change in the *Overhead MeOH Composition* for the *feedflow* data file, hold the left mouse down and highlight *Plot Var 1*. Release the mouse button. A graph of the transient response of the *Overhead MeOH Composition* should appear.

(e) It may be necessary to rescale the axes on the plot. This can be done using the *Trim Axes* option on the *View* menu of the graphics window (Hold your left mouse button on *View* and pull down as you have previously done to see this menu). You will be prompted to enter new values for the upper and lower limits for both the x-axis and the y-axis. **Hint: it may be helpful to set the x-axis lower limit equal to the time at which the change in the given input was introduced.**

(f) Once a plot of the desired output is obtained with the axes scaled as desired, a hard copy of the plot can be obtained using the *Print* option under the *File* menu (hold down the left mouse button and pull down underneath *File* on the graphics window. Highlight the *Print* option and release the left mouse button).

©2005 by PCS UTM

This will open the print options window. Selecting *OK* will print the current figure to the default printer.

4. Plot the *Overhead MeOH Composition* and the *Bottom MeOH Composition* for the *Vapour Flow Rate* change using *PCMplot* and the procedure described above. Fit a first-order-plus-time-delay transfer function model to each of the responses. Fill in your parameters below.

   Exercise 3 *Overhead MeOH Composition, G_{12}*
   Gain:

   Time Constant:

   Time Delay:

   Exercise 4 *Bottom MeOH Composition, G_{22}*

   Gain:

   Time Constant:

   Time Delay:

5. Plot the *Overhead MeOH Composition* and the *Bottom MeOH Composition* for the *Reflux Ratio* change using *PCMplot* and the procedure described above. Fit a first-order-plus-time-delay transfer function model to each of the responses. Fill in your parameters below.
   Exercise 5 *Overhead MeOH Composition, G_{11}*

   Gain:

   Time Constant:

   Time Delay:

   Exercise 6 *Bottom MeOH Composition, G_{21}*

   Gain:

   Time Constant:

   Time Delay:

6.4 Cohen-Coon Tuning Method

6.4.1 Introduction

Cohen-Coon method is a classical algorithm for PID controller tuning. This method yields controllers with the most a 0.25 value of the decay ratio (ratio of second peak to first peak in an

oscillatory response). While this leads, in principle, to an attenuation of oscillations, this technique typically leads to controllers with an aggressive response.

The Cohen-Coon tuning method requires an open-loop first-order-plus-time-delay transfer function model of the process. This can be obtained from a process reaction curve, generated from a step input. From the identified effective gain, time constant and dead time (K_p, τ_p, θ), one can tune the controller using the rules which are summarized below:

Controller Type	Controller Gain	Reset Time	Derivative Time Constant
P	$K_c = \left(\frac{1}{K_p}\right)\left(\frac{\tau_p}{\theta}\right)\left(1 + \frac{\theta}{3\tau_p}\right)$		
PI	$K_c = \left(\frac{1}{K_p}\right)\left(\frac{\tau_p}{\theta}\right)\left(\frac{9}{10} + \frac{\theta}{12\tau_p}\right)$	$\tau_I = \theta\left(\frac{30 + 3\theta/\tau_p}{9 + 20\vartheta/\tau_p}\right)$	
PID	$K_c = \left(\frac{1}{K_p}\right)\left(\frac{\tau_p}{\theta}\right)\left(\frac{4}{3} + \frac{\theta}{4\tau_p}\right)$	$\tau_I = \theta\left(\frac{32 + 6\theta/\tau_p}{13 + 8\vartheta/\tau_p}\right)$	$\tau_D = \vartheta\frac{4}{11 + 2\theta/\tau_P}$

Before you start the procedure, perform the following sequence of steps:

1. From the Distillation Menu click on the *Column with Control* button.

2. Make sure that both of the feedback loops are in manual by setting the loop switch blocks to the *off* position.

3. Start the simulation.

4. Make sure that the controller is properly tuned.

5. Allow the simulation to reach steady state before changing either of the feedback loops to automatic.

6.4.2 Procedure - Column

1. Tune the *Bottom MeOH Composition* controller with the proportional-integral (PI) controller tuning constants that you calculated from the Cohen-Coon tuning rules. Make sure that the integral action is *On* and the derivative action is *Off*. Introduce a new setpoint on the *Bottom MeOH Composition* of 0.1 mol MeOH/mol total.

   Exercise 7 Sketch the response.

2. Reset the *Bottom MeOH Composition* to its initial value, 0.15 mol MeOH/mol total. Once the system has returned to steady state, change the controller parameters to the PID settings obtained with the Cohen-Coon tuning rules for the *Bottom MeOH*

Composition. Make sure that integral action and derivative action are both *On*. Introduce a new setpoint on the *Bottom MeOH Composition* of 0.1 mol MeOH/mol total.

Exercise 8 Sketch the response.

Exercise 9 How does the PID-controlled response differ from the PI-controlled response?

3. Return the system to steady state by setting the *Bottom MeOH Composition Setpoint* to 0.15 mol MeOH/mol total. Tune the *Overhead MeOH Composition* controller with the proportional-integral (PI) controller tuning constants that you calculated from the Cohen-Coon tuning rules. Make sure that the integral action is *On* and the derivative action is *Off*. Introduce a new setpoint on the *Overhead MeOH Composition* of 0.9 mol MeOH/mol total.

   Exercise 10 Sketch the response.

4. Reset the *Overhead MeOH Composition* to its initial value, 0.85 mol MeOH/mol total. Once the system has returned to steady state, change the controller parameters to the PID settings obtained with the Cohen-Coon tuning rules for the *Overhead MeOH Composition*. Make sure that integral action and derivative action are both *On*. Introduce a new setpoint on the *Overhead MeOH Composition* of 0.9 mol MeOH/mol total.

   Exercise 11 Sketch the response.

   Exercise 12 How does the PID-controlled response differ from the PI-controlled response.

6.5 Summary

Closed-loop performance of control systems is greatly affected by the selection of controller tuning parameters. Trial and error methods to obtain controller tuning constants require many iterations. One of the methods is presented in this unit which provide good initial values for controller tuning.

©2005 by PCS UTM

Appendix

Section: Approved by : _____________

Group:

Date:

Exercise 1

Name of data file for this run __________________ (example: *vapourflow*).
Time at which the *Vapour Flow Rate* change is introduced _____________ sec (from clock display).

Exercise 2

Name of data file for this run __________________ (example: *reflux*).
Time at which the *Reflux Ratio* change is introduced _____________ sec (from clock display).

Exercise 3 *Overhead MeOH Composition,* G_{12}

Gain:

Time Constant:

Time Delay:

Exercise 4 *Bottom MeOH Composition,* G_{22}

Gain:

Time Constant:

Time Delay:

Exercise 5 *Overhead MeOH Composition,* G_{11}

Gain:

Time Constant:

Time Delay:

Exercise 6 *Bottom MeOH Composition,* G_{21}

Gain:

Time Constant:

Time Delay:

Exercise 7 Sketch the response**.**



Exercise 8 Sketch the response.



Exercise 9 How does the PID-controlled response differ from the PI-controlled response?

©2005 by PCS UTM

Exercise 10 Sketch the response**.**



Exercise 11 Sketch the response.



Exercise 12 How does the PID-controlled response differ from the PI-controlled response?

©2005 by PCS UTM

7

Programmable Logic Controller

7.1 Objective

Introduce the student to one of the most used Tools in control systems in the industry: the programmable logic controller.

7.2 Definition of PLC

1) An electronic device which receives inputs in form of digital or analog data and process them via a programmed sequence using timers, counters and mathematical elements to generate outputs to control a determined process or processes.
2) A class of industrial devices that perform logic functions that replace Electro-mechanical and solid state logic and other control functions.

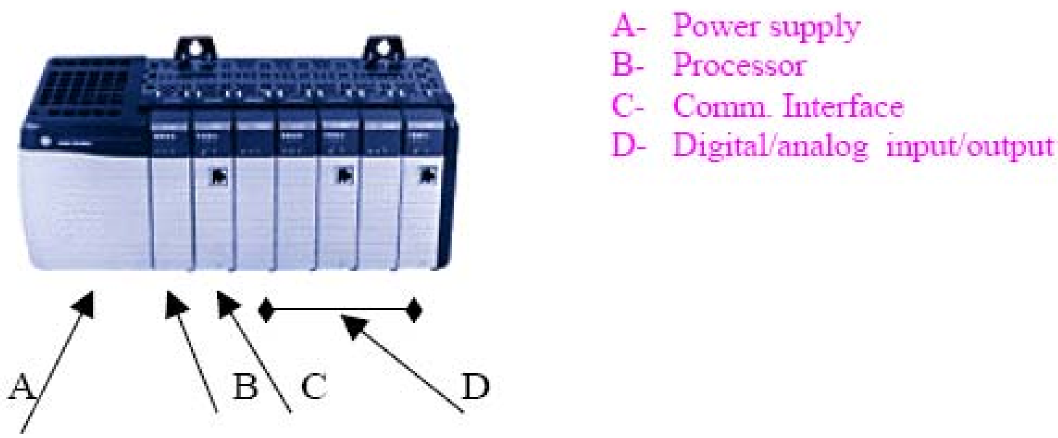
7.3 Classes of PLC’s

There are two classes of PLC’s: modular and compact.

3.1-Compact: It is the one that contains all the components of the PLC in one box. This type or class of PLC's are used in systems that do not have many inputs and outputs and can be seen in machinery automation and small processes.

3.2-Modular: It is the one that has all the elements (power supply, processor, inputs, outputs, etc…) separated by functions and are interconnected via a rack or base. This is the most widely used class in the industry and the one we will use.

7.4 Elements and Description



4.1-Power supply(A): It is the source of power to all the elements of the PLC. There are different types of power supplies and they are used depending on the requirements of the application.

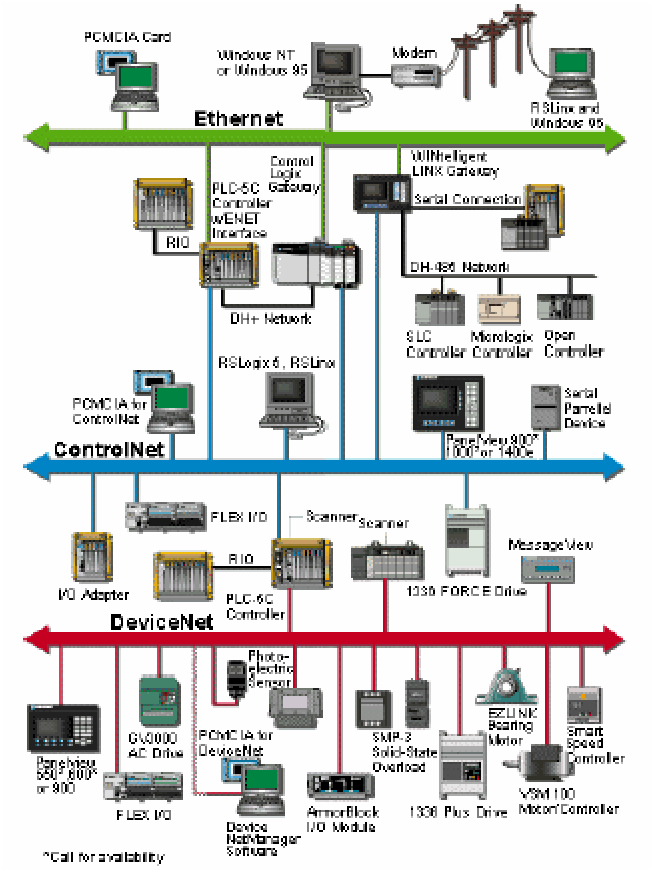
4.2-Processor unit: this unit is where all the processing is done. This unit receives all the information from the inputs, processes them and sends information to the outputs.

4.3- Inputs: They are the elements that receive information from the field or process and transfer them to the processor. That information can be analog or digital data.

4.4-Outputs: These are the blocks that establish the interface between the processor and the variables to be controlled in the process. Also they can be analog or digital.

4.5-Communication interface: There are many networks for the SLC (we will focus on the Controlnet) and so are interfaces. We will use the dh-485 interface.

©2005 by PCS UTM



Communication card: The communication card of the PLC can be connected to a network or a computer. That makes the job easier in terms of programming and debugging the system. The protocol to use is the dh-485 in a controlnet network.

There are two types of cards: the “personal computer interface” and the “dh-485 interface”.

1746-PIC: The personal computer interface is used to connect one PLC to the computer via the serial port. This configuration is widely used in *on site programming and debugging* with a laptop.

1747-AIC: The dh-485 interface is connected between the PLC and the computer interface. This converter is used to connect various PLC in a network and to a computer, so now the computer can interact with more than one PLC via the same port.

7.5 Sensor and actuators:

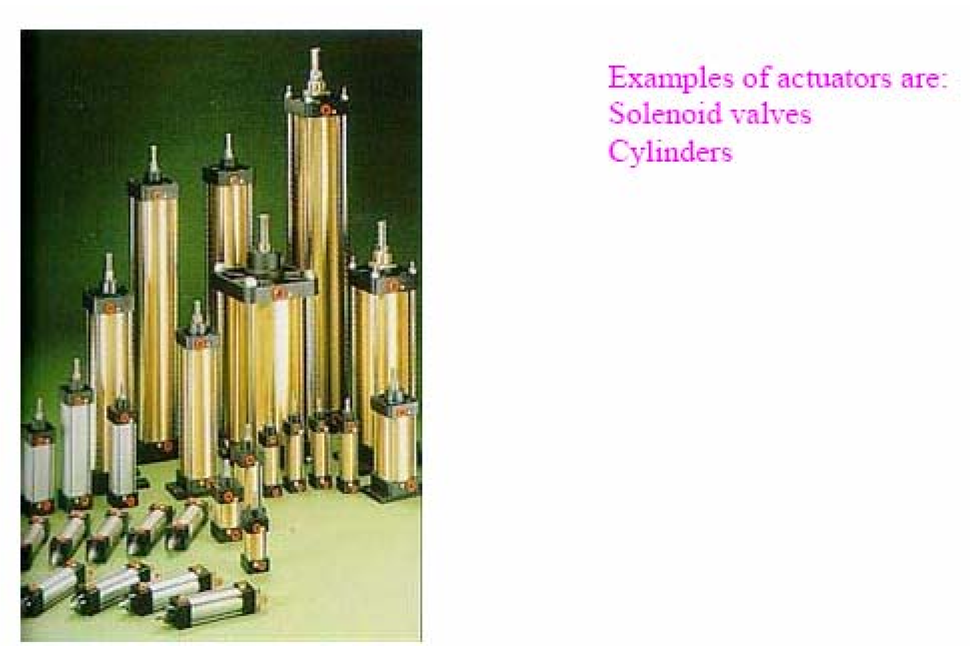
5.1-Sensors: They are used to get information from the process. These elements are connected to the inputs of the PLC. There are different types of sensors:

1) Proximity (a) (capacitive and inductive).

2) Temperature (thermoresistor (b), temperature gauge(c), pyrometer, thermocouple (d)).
3) Pressure (pressure Gauge, piezoelectric valve).
4) Photoelectric sensors

All these elements are used to receive information from the controlled process. Some of them generate digital signals (a) and are connected to the digital inputs of the PLC, other generate analog signals (b, c, d) and are connected to the analog inputs.

5.2-Actuators: They are connected to the outputs of the PLC that control the components of the system to adjust them to the set point.



5.3-User interface: It is used to monitor the process through information given in the PLC.



7.6 Uses of the PLC:

Robotics, industrial automation, hazard environment control, mining, etc…

7.7 Software: Recall of digital electronics:

The bases of the digital world are the gates. The systems are composed of them in different arrangements and quantities. Here the goal is to compare the gate circuits with the ladder circuit or ladder logic.

Comparing gates and ladder logic: The gate system is used in digital electronics and the ladder logic is used more in electrical systems, this is a simple relation in a one by one basis. Assume three digital variables A, B and C. A and B are the inputs and C is the output. The table below shows the response of the different types of gates and ladder circuits.

And gate:

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



Or gate

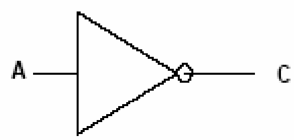
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

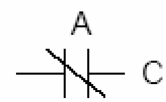


©2005 by PCS UTM

Not gate

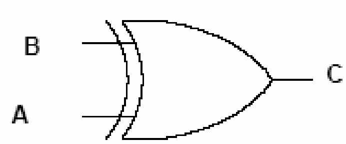
A	C
0	1
1	0

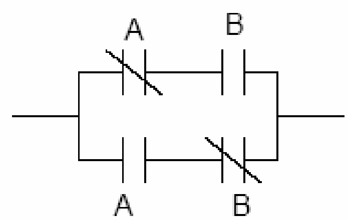




Xor gate

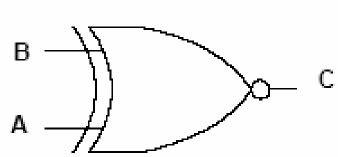
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

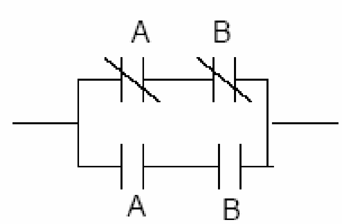




Xnor gate

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1





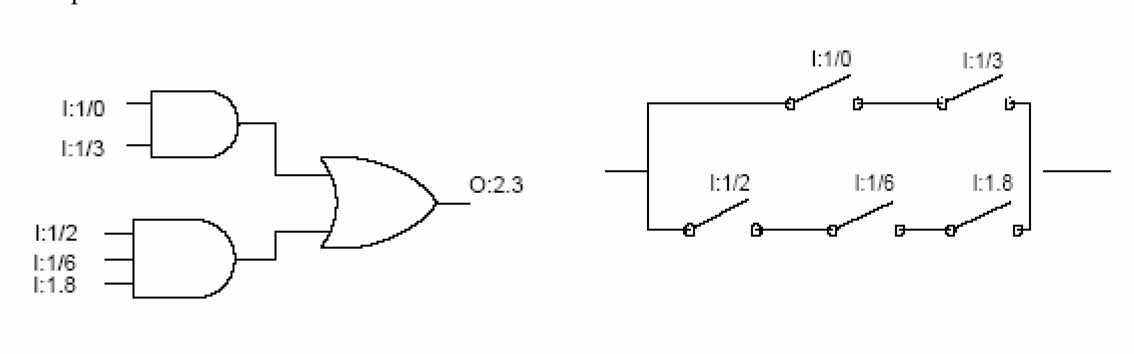
©2005 by PCS UTM

Think about the ladder logic as a group of switches, ones normally open and other normally closed. Vertical parallel lines represent the switches. Generally they are normally open; the normally closed have a diagonal line between the parallel lines.

The normal position of the switch is the one that they will get when not active, when activated that position changes to open or close depending on the type of switch. The PLC recognizes logic by ladder so the systems have to be converted to that type of representation.

Tips: When simplifying logic circuits, use normal logic and then transfer the result to ladder.

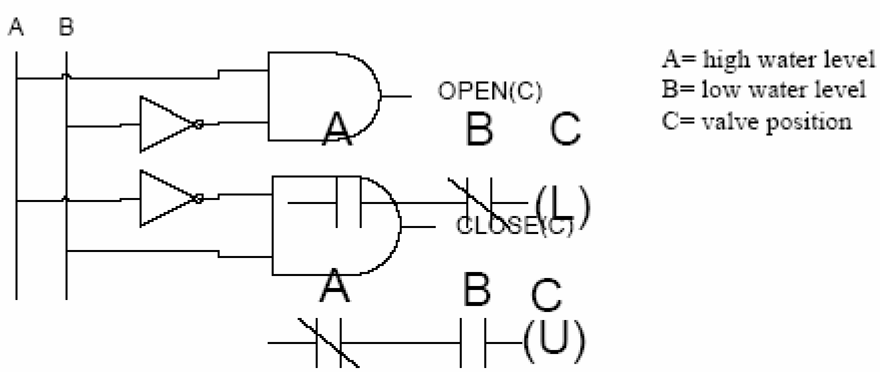
Example #1:



Example #2:

Let the process be the level of a water tank. There are two sensors: one at low level and the other at high level. The variable to control is the level of the tank. There is a faucet at the top of the tank that fills it up and other pipe where the water goes out. The quantity of water flowing through the exit pipe is random and is not controlled by the process. The control system has to maintain the water level between the low and high level presetted.

The logic for this problem is like this:



There are (2) inverters and (2) AND gates in the circuit. The ladder is seen like this:

8

Dynamic Simulation in Simulink

8.1 Objective

The purpose of this module is to introduce SIMULINK, a block diagram environment for simulation. After studying this module, you should be able to construct a block diagram for simulation and set proper simulation parameters.

8.2 Introduction

There are limitations to using the standard MATLAB environment for simulation. Engineers (and particularly control engineers) tend to think in terms of block diagrams. Although MATLAB functions such as *parallel*, *series*, and *feedback* allow the simulation of block diagrams, these functions are not visually pleasing. SIMULINK provides a much more natural environment (a graphical user interface, GUI) for simulating systems that are described by block diagrams. SIMULINK also provides integration methods that can handle systems with time-delays (rather than making a Padé approximation for deadtime).

©2005 by PCS UTM

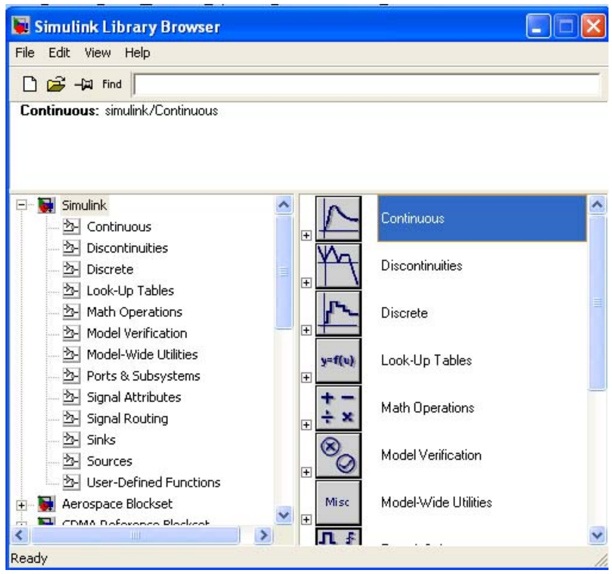


Figure 8.1: SIMULINK library browser

We will illustrate the use of SIMULINK by way of example.

In the MATLAB command window, enter "simulink" (small letters):

>> simulink

The SIMULINK block library window appears, as shown in Figure 8.1.

There are many possible SIMULINK functions (icons) available. Double-clicking on the *Simulink Extras*, you will find some additional blocks you may use in your simulation. The resulting window is shown in Figure 8.2.

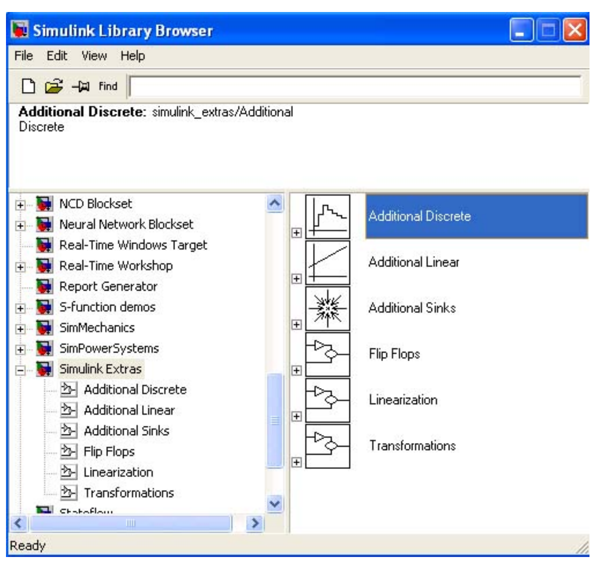


Figure 8.2: SIMULINK *Extras* block

8.3 Transfer Function-Based Simulation

Here we provide an example simulation of a first-order process. The first step is to create a new model by clicking *File – New – Model* in the SIMULINK library browser. An untitled SIMULINK model will appear as shown in Figure 8.3.

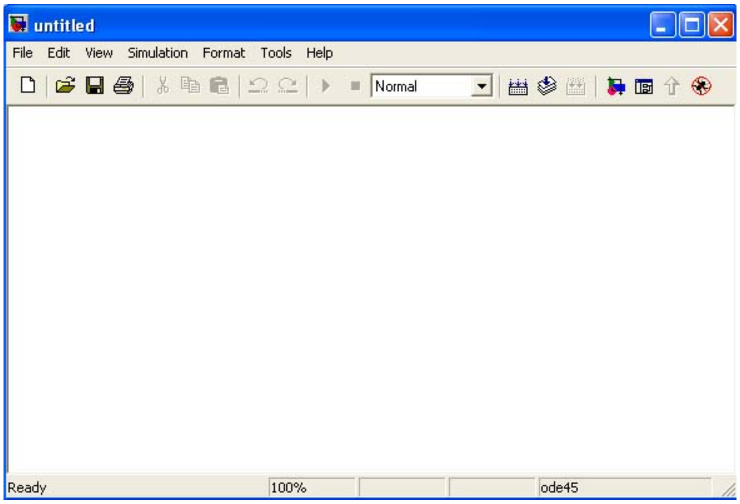


Figure 8.3: Untitled SIMULINK model

Construct a SIMULINK simulation with the following elements:

(a) Clock

(b) Step input

(c) Transfer function

(d) To Workspace (2 unit)

Drag icons from the SIMULINK library browser to the Untitled SIMULINK model. The *clock* is used to create a time vector. The *To workspace* icons allow the vectors to be written to the MATLAB workspace for later manipulation or plotting.

Notice that each of the variable names, as well as the icon names, can be changed by the user, as shown in Figure 8.4. The variable names are changed by double-clicking on the icon and changing the name in the variable space.

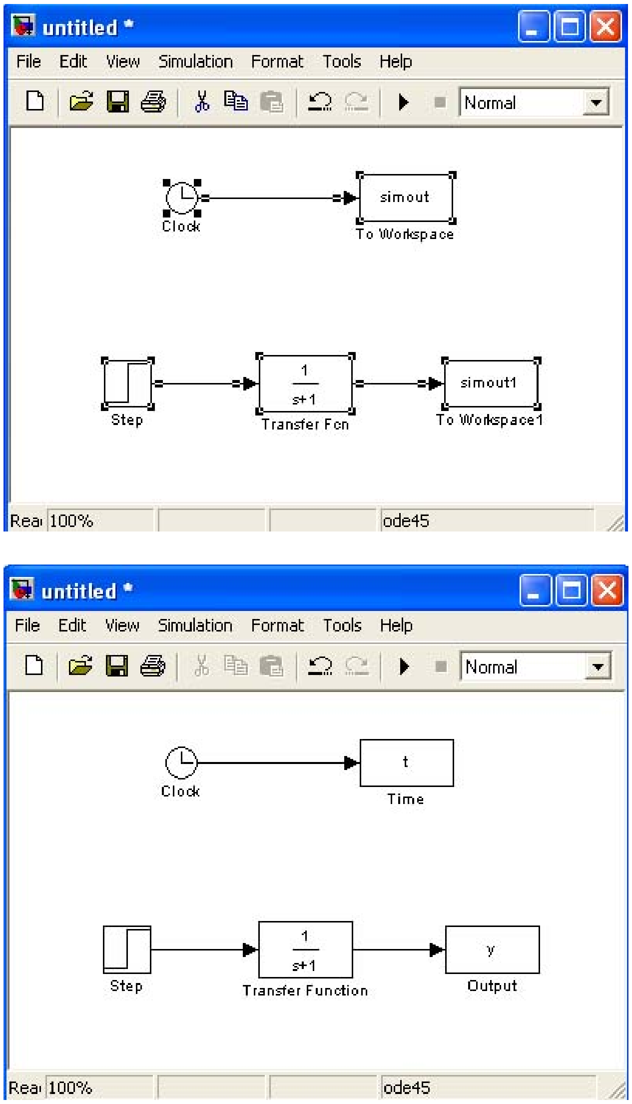


Figure 8.4: Blocks for transfer function simulation (top) and blocks for transfer function simulation – variables and icons renamed.

Notice that the default transfer function has a pole at zero, that is, as simple integrator. This is changed by double-clicking on the icon and entering numerator and denominator coefficients. An example of a system with a gain of 2 and a time constant of 5 is shown in Figure 8.5.

The default step input is to step the input from 0 to 1 at t = 1. This can easily be changed by double-clicking on the *Step* icon and changing the specifications.

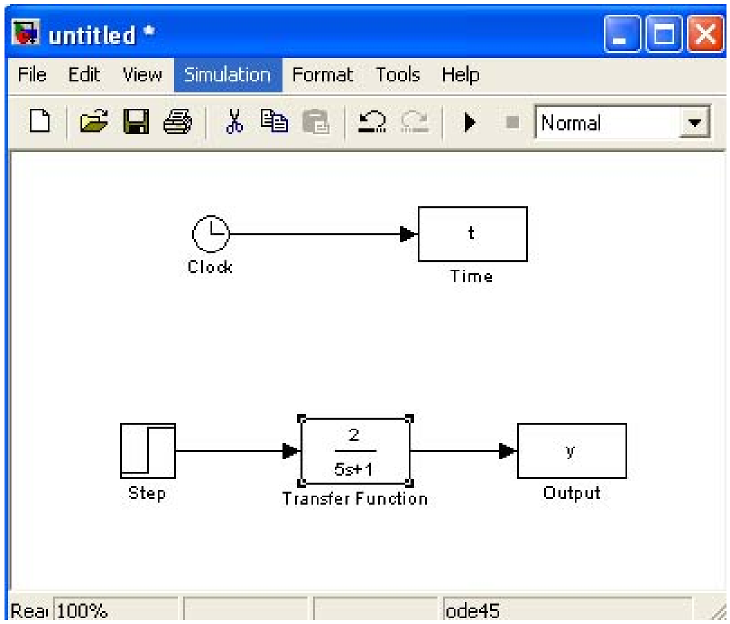


Figure 8.5: Transfer function simulation – first-order process

The simulation parameters (start time, stop time, integration method, etc.) are changed by using the simulation pulldown menu and selecting *Simulation Parameters*. Change the *Stop* Time to 25 and *Max Step* Size to 1. The default integration method is ODE45 (Runge-Kutta). You may wish to try several different integration methods, depending on the type of system you are attempting to simulate.

Before start the simulation, make sure to change *Save Format* in the *To Workspace* from *Structure* to *Array*. This is to make the vectors to be written to the MATLAB workspace in array form. To begin the simulation, simply select *Start* from the simulation pulldown menu. The *t* and *y* variables are stored in the MATLAB workspace. The resulting response is shown in Figure 8.6. Notice that the input was stepped from 0 to 1 at $t = 1$, that is, the default values were used.

A major advantage of SIMULINK over the standard MATLAB integration routines is the ability to handle systems with time delays. This is done using the *Transport Delay* block shown in Figure 8.7. Here we use a time-delay of 5, obtained by modifying the default value of 1. It should also be noted that the user must apply an initial input value for the transport delay block.

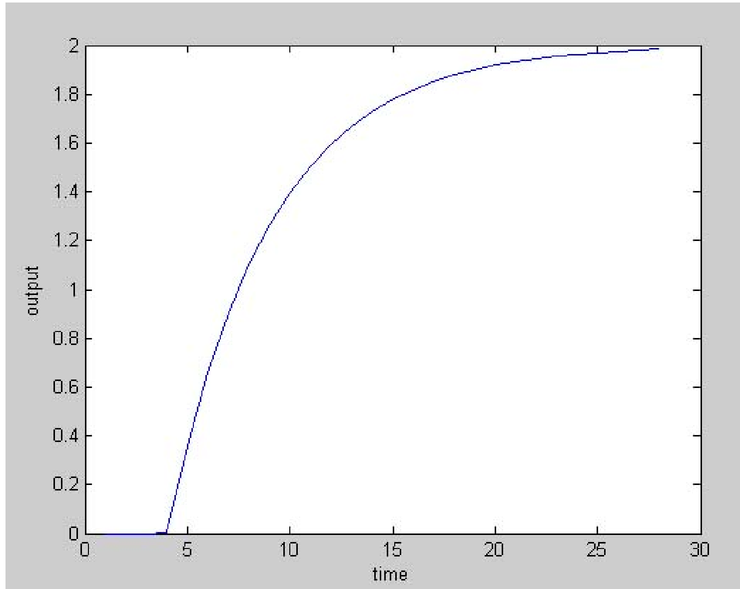


Figure 8.6: Simulation result for first-order example with a step input at $t = 1$

The resulting simulation is shown in Figure 8.8. Notice that the output changes after 5 time from the previous simulation. This is because the step input changing after 5 unit time delay.

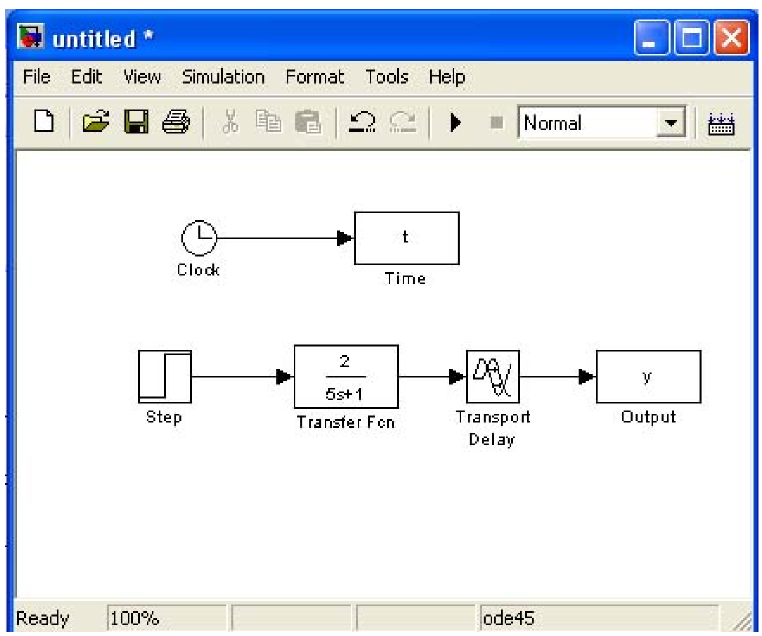


Figure 8.7: Transfer function simulation – first-order + time delay process

©2005 by PCS UTM

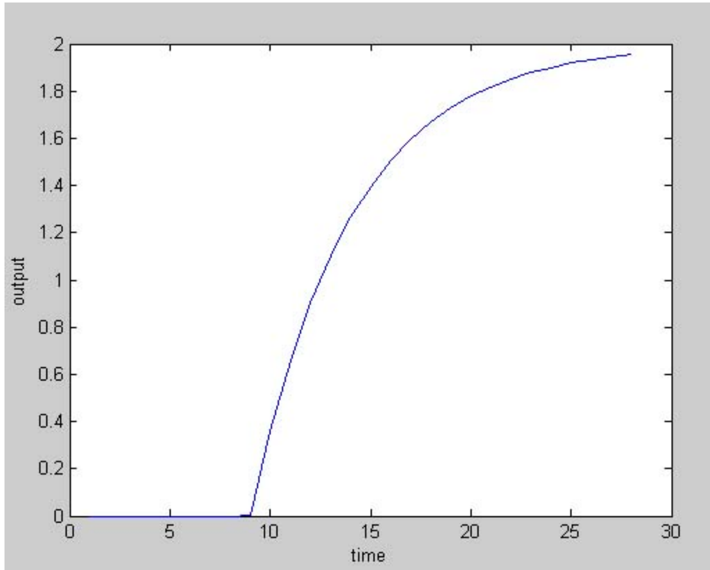


Figure 8.8: Simulation result. Step input for first-order system with time delay of 5

Although the block diagram feature of SIMULINK is very handy, it would be time consuming (and tiring) if every time you wanted to change a set of parameters you had to double-click on the related icons. You can place variable names, such as *k*, *tau*, *theta*, for a first-order + time delay system directly in the blocks, the simply change the values (*k*, *tau*, *theta*) in the MATLAB command window before performing a new simulation. This block diagram is shown in Figure 8.9.

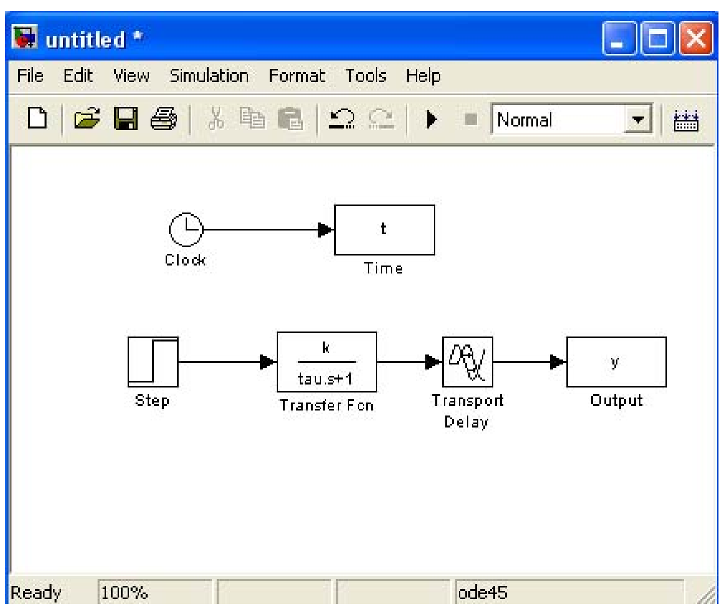


Figure 8.9: Transfer function simulation. The values of *k*, *tau*, and *theta* can be entered in the MATLAB command window before beginning the simulation

Use SIMULINK to perform the following. Show the SIMULINK diagram used for your simulations. Set *Stop Time* to 100 and *Max Step Size* to 1.

Exercise 1 Plot the step response of the following underdamped system

$$g(s) = \frac{3}{25s^2 + 5s + 1}$$

when a step input of magnitude 2 is made at t = 1. Comment the response.

Exercise 2 Plot the step response of the following underdamped system

$$g(s) = \frac{3(-3s + 1)}{25s^2 + 5s + 1}$$

when a step input of magnitude 2 is made at t = 1. Comment the response.

Exercise 3 Plot the step response of the following underdamped system

$$g(s) = \frac{3(-3 + 1)e^{-5s}}{25s^2 + 5s + 1}$$

when a step input of magnitude 2 is made at t = 1. Comment the response.

8.4 SUMMARY

The purpose if this module was to provide a brief introduction to the use of SIMULINK for block diagram programming. SIMULINK provides a much more natural environment (a graphical user interface, GUI) for simulating systems that are described by block diagrams. It also provides methods that can handle system with time-delays.

©2005 by PCS UTM

8.5 APPENDIX

Section: Approved by: _____________

Group:

Date:

Exercise 1

SIMULINK Block diagram

Response

Comment:

Exercise 2

SIMULINK Block diagram

Response

Comment:

©2005 by PCS UTM

Exercise 3

SIMULINK Block diagram



Response

Comment: